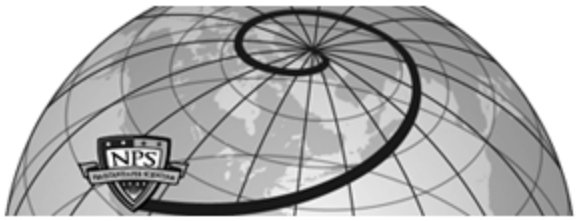


CONTROL SYSTEM PROGRAMMING REMOTE COMPUTING AND DATA DISPLAY

ROBERT LEE HOGG
and
DENNIS C. GLOVER

24

CONTROL SYSTEM PROGRAMMING
REMOTE COMPUTING
and
DATA DISPLAY

by
Robert Lee Hogg
Lieutenant, United States Navy
and
Dennis C. Glover
Lieutenant, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of
MASTER OF SCIENCE
IN
ENGINEERING ELECTRONICS
United States Naval Postgraduate School
Monterey, California
1 9 6 3

CONTROL SYSTEM PROGRAMMING
REMOTE COMPUTING
and
DATA DISPLAY

****

Robert Lee Hogg
and
Dennis C. Glover

CONTROL SYSTEM PROGRAMMING
REMOTE COMPUTING
and
DATA DISPLAY
by
Robert Lee Hogg
and
Dennis C. Glover

This work is accepted as fulfilling the
thesis requirement for the degree of
MASTER OF SCIENCE
IN
ENGINEERING ELECTRONICS
from the
United States Naval Postgraduate School

ABSTRACT

The increasing demands upon large digital computing facilities have necessitated the developement and use of a system control program to provide for continuous automatic job processing. A comparative study of two current systems was conducted. The first, CDC FORTRAN 60, is described and its advantages and disadvantages are pointed out. This system was modified by the authors to incorporate a remote (satellite) operation in a time shared mode. The second system studied was the CDC COOP MONITOR system. This system, a more complex, sophisticated control system is likewise discussed. Modifications to correct certain disadvantages of this system are shown, along with the programming necessary to provide for remote station operation in the COOP MONITOR environment.

Several other programs which were developed by the authors to improve the computing center's capability are documented. These include; (1) A graph plotting routine utilizing CDC 1604/160 computors and a CDC 165 / CalComp Plotter, (2) A large file merg sort routine, and (3) A data display "Line Printer" routine.

The authors wish to express their appreciation to Professor Mitchell L. Cotton of the U.S. Naval Postgraduate School, and to the personnel of the school's computing center for their kind consideration during the term of this thesis.

TABLE OF CONTENTS

APPENDICES

LIST OF ILLUSTRATIONS

ENCLOSURES

ABBREVIATIONS

MCS	Master Control System
SCS	Secondary Control System
RHT	Running Hardware Table
SIO	Standard Input Output
AET	Available Equipment Table
AEDNT	Available Equipment Driver Name Table
RBL	Relocatable Binary Loader
ELT	Edit Library Tape
PLT	Prepare Library Tape
FWA	First Word Address
LWA+1	Last Word Address Plus One
I/O	Input/Output
BCD	Binary Coded Decimal
CDC	Control Data Corparation

1.0 INTRODUCTION

At present, the Naval Postgraduate School's computing center utilizes approximately 60% of its prime time processing short, 1 to 15 minute, "job shop", input programs. Practically all of these are of a scientific problem solving nature. Three different compiler/assembler control systems for processing these programs are currently in significant usage. They are the Control Data Corporation's "Fortran 60 System", the Naval Electronics Laboratory's "Neliac Compiler", and the school's own algebraic assembly routine, "Scrap". Of the above, the Fortran 60 System is the only system which incorporates a full control, automatic, job sequencing "monitor" system. Much more will be said about this "monitor" capability and the Fortran 60 System. The remaining 40% of the prime computing time is utilized for class laboratory instruction, maintenance and update of systems library tapes, data retrieval, schedule analysis, and other special job users.

The non-prime time, 1630 to 0800, is scheduled block time for the various developement projects and long run programs.

This environment places increasing demands upon the computing center's capabilites. Not only is improved efficiency required but also more flexibility is in demand as the programmer's talents become more and more sophisticated.

Several tasks were undertaken in the scope of this thesis project. Towards improving over-all system efficiency, a comparative study of the present Fortran 60 System and its proposed successor, "Coop Monitor/Fortran 62 System", was conducted. As a result, an outline and a discussion of the outstanding advantages and disadvantages of each system is incorporated in this thesis.

Towards improved system flexibility, two new facilities were developed. They were the implementation of an operational remote (satellite) computing station and a system's provision for automatic graph plotting of two dimensional computer

derived arrays.

The full implementation of the remote station complex is a continuing group project with much remaining to be accomplished. As an aid to those who pursue this task, several recommendations are made and suggested methods for solving some of the problems are expounded.

Besides the work accomplished in establishing the satellite station, another aspect of the center's services was improved. This was the provision of a graph plotting system. This required the developement of a plotting program for both the Fortran 60 System and a much improved and shorter Grafplot program for the Coop Monitor System. The major portion of this project was completed during the author's five week industrial tour at the Control Data Corporation, Palo Alto, California. The off-line processing of this graph plotting program was provided by rewriting, with many revisions, an existing plot drive program initially developed by the U.S. Naval Fleet Numerical Weather Facility, Monterey, California. Versions of this program were developed to operate on either the 160 or 160A CDC computers. The plotter employed is the CDC model 165/CALCOMP incremental plotter. This system has proven very effective and has enjoyed considerable usage. Although this provided an improved data output medium for the center, it has also stimulated an increased interest in utilizing the Digital Computer for project studies. Thus the center's task and effectiveness is ever increasing.

A second method of data display was provided by utilizing a new cathode ray tube, Data Display model DD65, as an "on-line" line printer. Thus it is possible to monitor the Fortran System's output listings concurrently as it is listed upon the magnetic tapes for later off-line hard copy printing.

This is a small programmed package to demonstrate a unique method of providing a "rolling page" effect on the display tube. The details of the technique are explained in section 4. It was originally planned to provide much more programming for the data display unit, however, the unexpected complexity of the Coop Monitor System limited the extent to which the authors were able to pursue this task. Modifications to the Coop Monitor System were made to couple the data display to the system. If time permits before termination of this work, the display driver routines will be completed to provide normal input/output capability between the display and the central processing computer.

Two additional projects of secondary importance were completed. They were the provision of tape control programs to enable utilization of the 1607 magnetic tape units by the CDC 160 computers when assembling programs using the CDC 160 OSAS Assembly System. The other project was the developement of a merg sort routine for large files written in 120 character BCD format. This sorting capability was a very desireable ability which had not previously been available at this installation.

2.0 FORTRAN 60 SYSTEM

The Fortran 60 System is the present fortran processor in use at the Postgraduate School. It is maintained on a system library tape, recorded in binary format, 54 words per record, see ref.21. The records are grouped sequentually such that one record or several records may constitute a program package. These packages are referred to as being "subroutines". The first such package on the system library tape is the System Control Routine and is titled "Resident". Operation of the system is as illustrated by figure 1. The figure points out the control transfer sequence which takes place after the resident program is "bootstrapped" into the 1604 computer's memory. A short verbal explanation of the diagram follows: Initial "bootstrap" procedures are executed by the operator. This reads in the Resident program from the system library. The resident program then assumes control of the computer.

2.1 Program Control

The section of resident in which control normally resides is termed "Program Control". The function of program control is to interpret control instructions which will direct the transfer of control to some secondary control package or subroutine. All such routines must eventually return control to "Program Control". Normally program control reads its instructions from the console typewriter. These control statements may be any one of the twenty routine names shown by figure 1, or one of the eight transient program names which previously may have been "called" into core (memory) by the call routine. Some of the twenty control statements have arguments. The CDC Fortran System Manual and its update revision notices

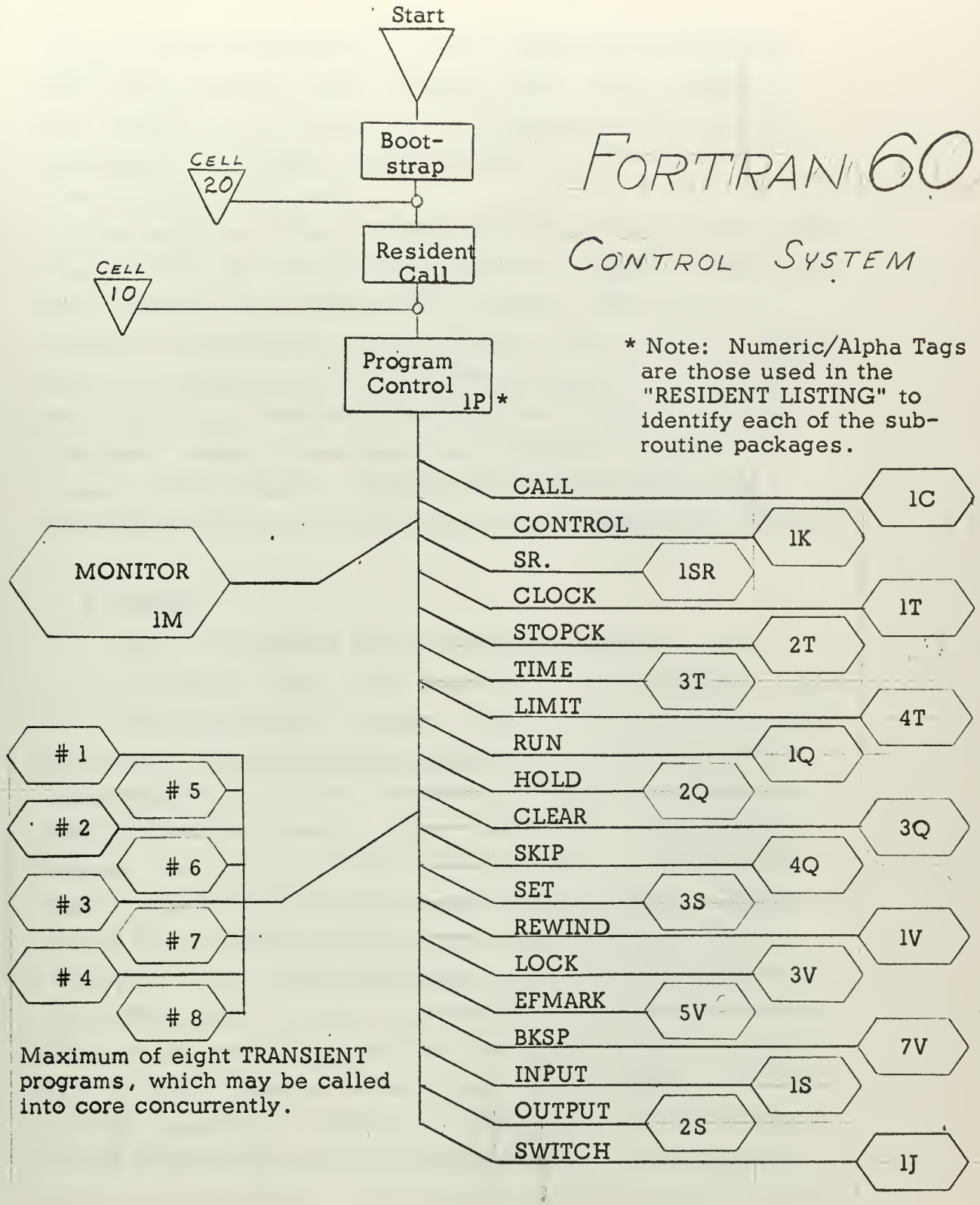

Start
Boot-strap
CELL 20
Resident Call
CELL 10
Program Control 1P *
FORTRAN 60
CONTROL SYSTEM
* Note: Numeric/Alpha Tags are those used in the "RESIDENT LISTING" to identify each of the sub-routine packages.
MONITOR 1M
CALL 1C
CONTROL 1K
SR. 1SR
CLOCK 1T
STOPCK 2T
TIME 3T
LIMIT 4T
RUN 1Q
HOLD 2Q
CLEAR 3Q
SKIP 4Q
SET 3S
REWIND 1V
LOCK 3V
EFMARK 5V
BKSP 7V
INPUT 1S
OUTPUT 2S
SWITCH 1J
1
2
3
4
5
6
7
8
Maximum of eight TRANSIENT programs, which may be called into core concurrently.

Figure 1

discuss the use of these arguments and control statements, with the exception of the statement "SR". This statement was added by the authors to provide automatic call of eight frequently used service routines such as transfer, vfylib, list, copys, verify, etc.

As mentioned before, normal operation of the control system requires that all sub-program packages, to which control may be transferred, must provide for returning control back to Program Control at the end of the directed processing sequence. Thus an examination of the resident programming, appendix IV, will find that it consists of several small programming sub-routines. Some of these are self contained and some are linked to other routines which assist in the performance of the functional task for which each basic package was designed.

2.2 Monitor

It has been pointed out that "Program Control" has a very responsible task in the organization of the system, since it is the prime control routine. The next most important and the largest ofthe programming packages in resident is the processor for the control statement, "Monitor" or sometimes called "Monitor Control". It is a routine to which program control can transfer control of the computer to provide automatic processing of stacked input Fortran Programs. Monitor is by far the most complex of the resident routines and yet, compared to the second generation system, Coop Monitor, it is rather simple. A full understanding of the monitor routine is best obtained by reading this section of coding in appendix IV. Briefly, operation of the monitor is as follows. Program Control receives a statement, "Monitor,3,4." This statement could have had up to seven arguments, but for this example two will be sufficient. This statement directs program control

to transfer the arguments into the monitor control routine and then to transfer control to monitor. Monitor then tests certain flags to check if recovering from a previous job execution which lost control, requiring a forced recovery entry back to monitor. This not being the case, monitor then rewinds all output mediums, senses for input/output conflicts, initilizes for next library routine call, builds a tape assignment table, reads from the input medium (tape 3 in this example), detects a job description record, calls in the Fortran compiler, inserts the necessary arguments into the compiler, then transfers control to the compiler for compilation of the input program. If the program is successfully compiled, control is returned to monitor and monitor then transfers control to the program just compiled, this is termed "executing the program". If the program is successfully executed, it returns control back to the monitor which then returns to process the next job on the stacked input medium.

2.3 Resident Routines

The technique of transferring control of the computer has been demonstrated and we should now discuss a technique of utilizing the resident routines which are available to non-resident programs. The organization of the system provides what are called "low core" (cells 7 thru 35), entry points. These entry points are assigned to be the permanent entries to several of the programming packages contained in resident. All of these entries are shown in figure 2. Since these cells are permanently assigned, standardization limits all usage of the resident routines by non-resident programs to enter via the appropriate low-core cell. This then allows modifications to be made on the system's "Resident" routine without influencing the non-resident routines. As illustrated by

figure 2, eight of the low-core entry routines have switches which condition on whether monitor is controlling the job sequencing. If the monitor is in effect, then these eight entries are diverted in order to trap illegal monitor usage of input/output units.

The low-core entry routines are rather self explainatory. They are mostly service routines such as read, write, equipment controllers, and message handlers.

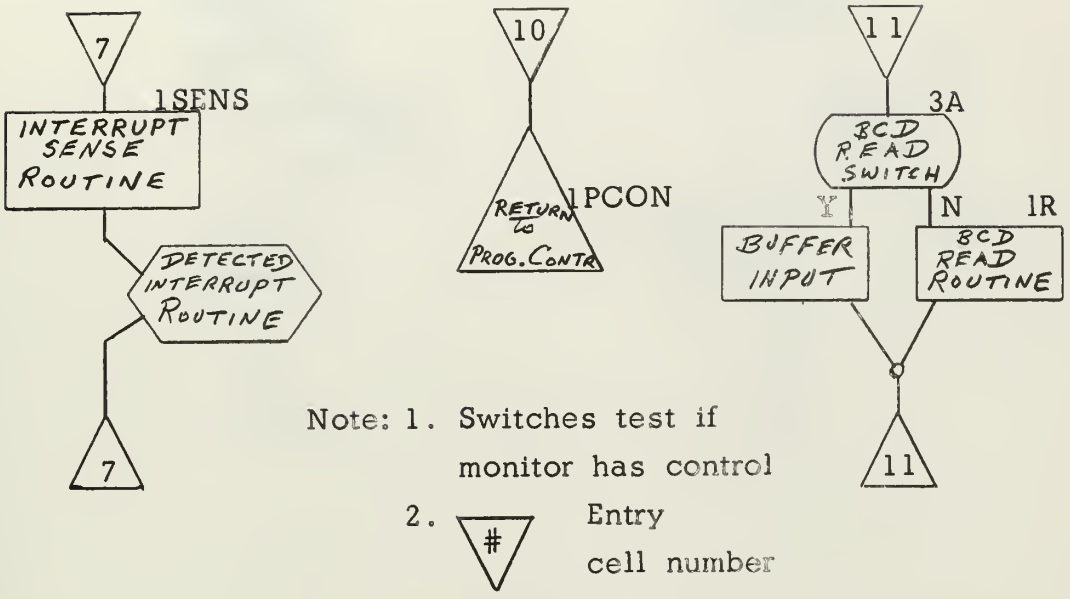
Nearly all the routines in resident fall into one of the two groups discussed. There are a few which are not in either group. These shall now be discussed.

Two similar routines which have not been mentioned are the "Input Index Register Loader" and the "Output Index Register Loader" routines. These two serve to translate the alphabetical equipment designators into the equipment codes as listed on the first page of the "Resident List", appendix IV. These equipment codes are then loaded into index registers #1 and #2. The index registers then control the proper equipment selection for the read/write routines.

Two other routines are the "Tape Read/Write" routines. These are both tape drive routines for controlling the system's magnetic tape units. The routines are commoly used by both the BCD and Binary Read/Write routines. The tape drive routines apparently were the only equipment drivers deemed worthy of individual programming. All other equipment drivers are embedded into the main body of the Read/Write routines.

There is a routine called, "Initialize for Next Call". Its function is to reset the program bias and program counter cells, #62 and #63 respectively, prior to calling each program from the library tape. This routine is bypassed if

"LOW CORE", CELLS 7 THRU 33
RESIDENT ROUTINE ENTRIES
WHICH ARE AVAILABLE TO
NON-RESIDENT PROGRAMS



Note: 1. Switches test if monitor has control

2. Entry cell number

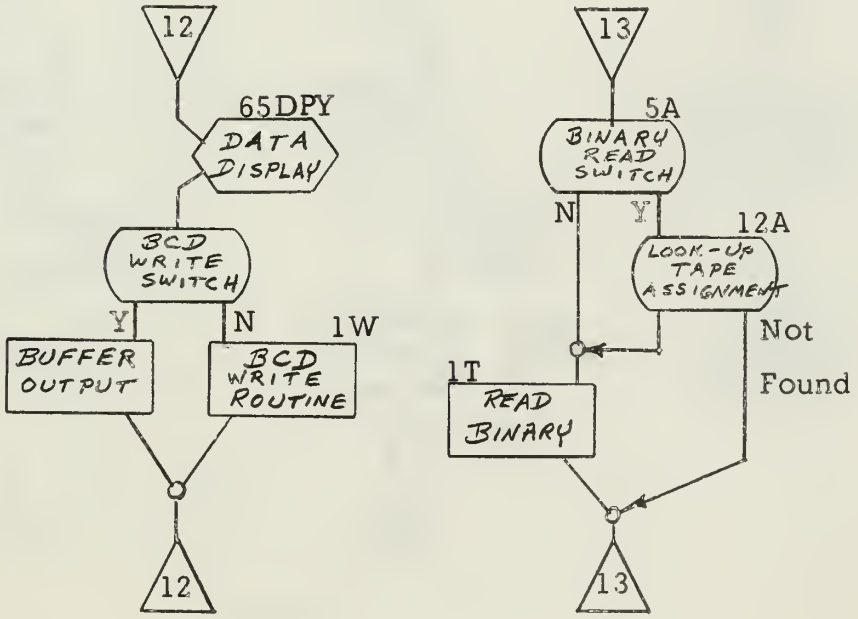


Figure 2 a

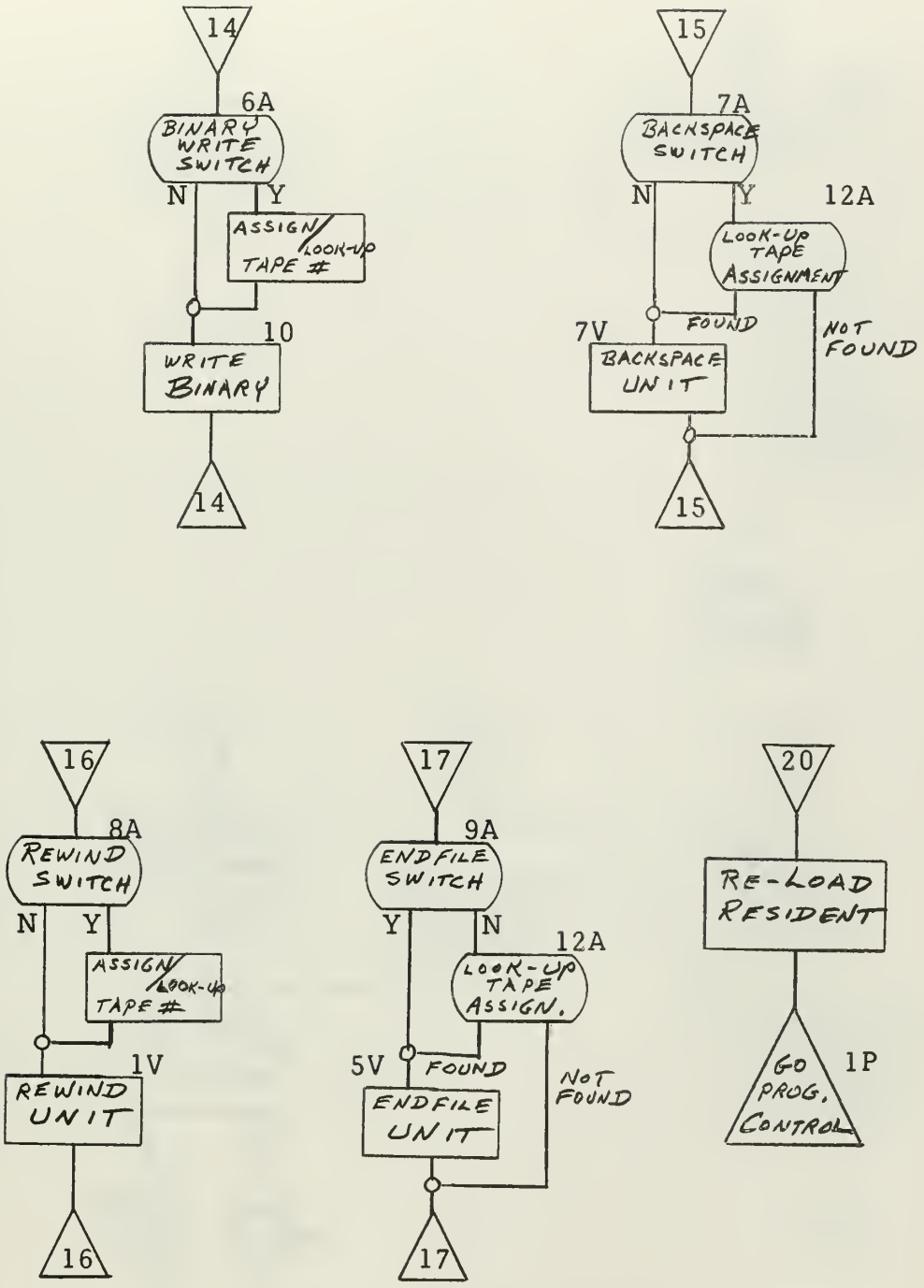

14
6A
BINARY WRITE SWITCH
N
Y
ASSIGN/ LOOK-UP TAPE #
10
WRITE BINARY
14
15
7A
BACKSPACE SWITCH
N
Y
12A
LOOK-UP TAPE ASSIGNMENT
FOUND
NOT FOUND
7V
BACKSPACE UNIT
15
16
8A
REWIND SWITCH
N
Y
ASSIGN/ LOOK-UP TAPE #
1V
REWIND UNIT
16
17
9A
ENDFILE SWITCH
Y
N
12A
LOOK-UP TAPE ASSIGN.
5V
FOUND
NOT FOUND
ENDFILE UNIT
17
20
RE-LOAD RESIDENT
GO PROG. CONTROL
1P

Figure 2b

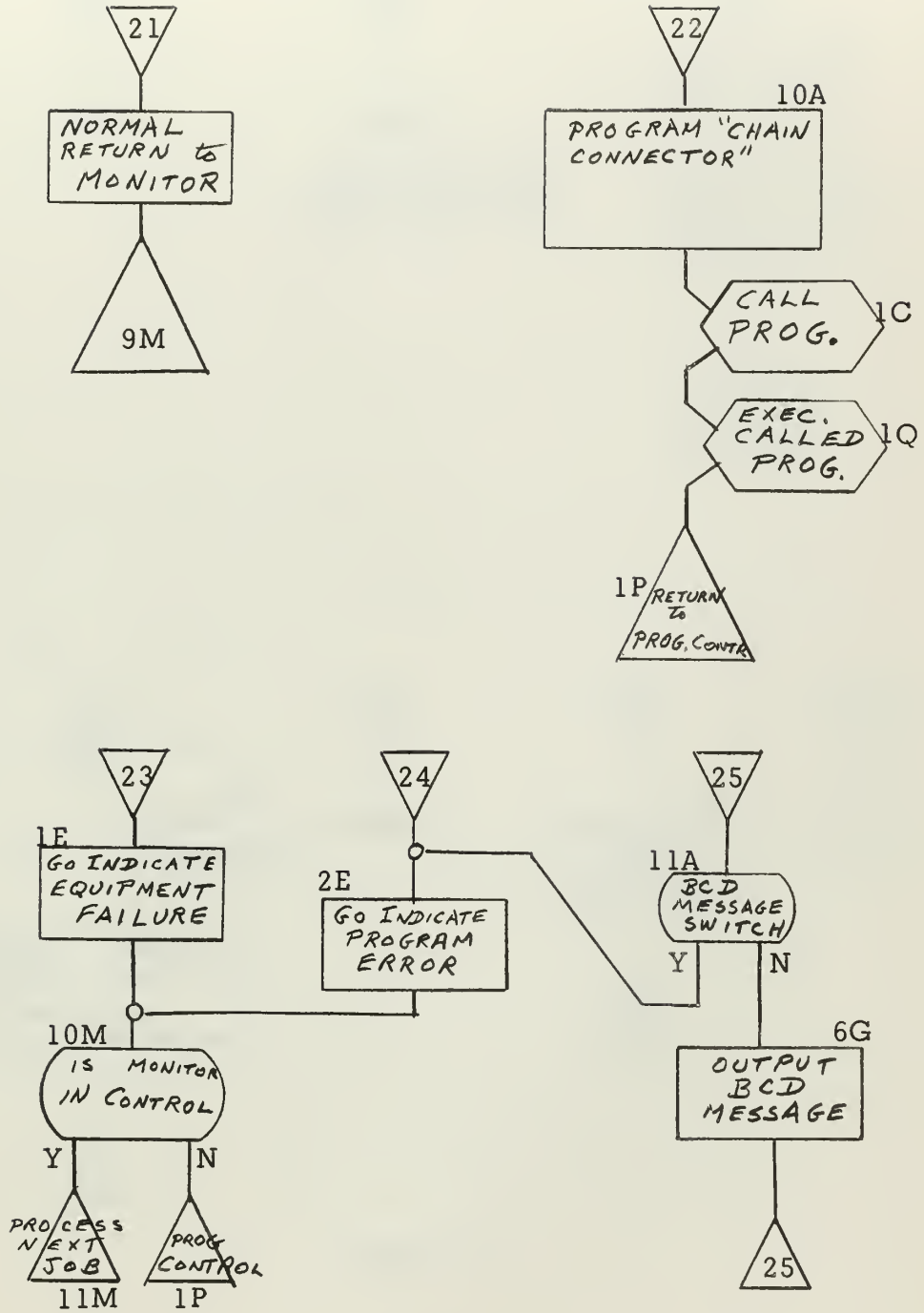

21
NORMAL RETURN to MONITOR
9M
22
10A
PROGRAM "CHAIN CONNECTOR"
CALL PROG.
1C
EXEC. CALLED PROG.
1Q
1P
RETURN To PROG. CONTR.
23
1E
GO INDICATE EQUIPMENT FAILURE
24
2E
GO INDICATE PROGRAM ERROR
25
11A
BCD MESSAGE SWITCH
Y
N
10M
IS MONITOR IN CONTROL
Y
N
6G
OUTPUT BCD MESSAGE
PROCESS NEXT JOB
11M
PROG CONTROL
1P
25

Figure 2c

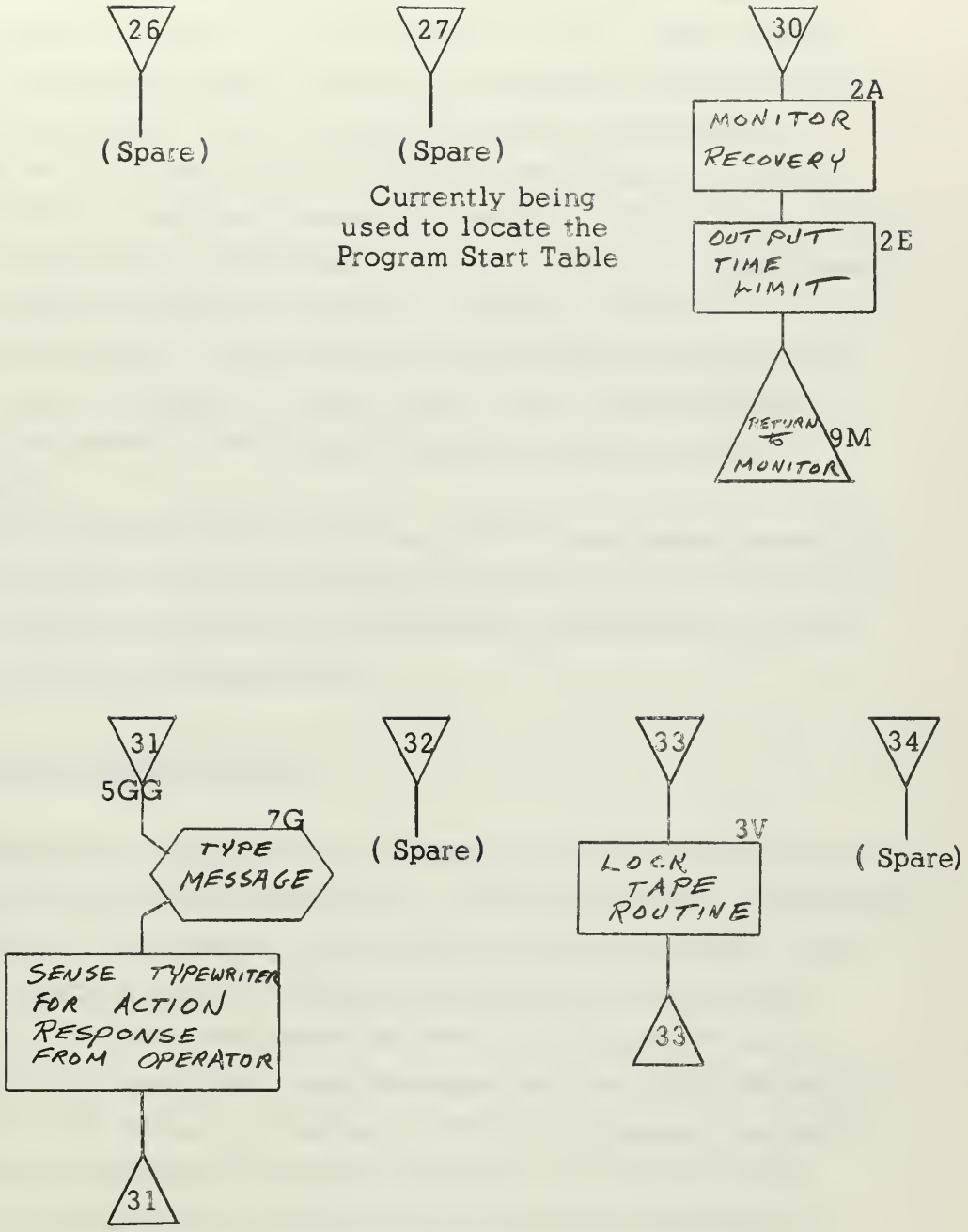

26
(Spare)
27
(Spare)
Currently being used to locate the Program Start Table
30
2A
MONITOR RECOVERY
OUT PUT TIME LIMIT
2E
RETURN TO MONITOR
9M
31
5GG
7G
TYPE MESSAGE
SENSE TYPEWRITER FOR ACTION RESPONSE FROM OPERATOR
31
32
(Spare)
33
3V
LOCK TAPE ROUTINE
33
34
(Spare)

Figure 2d

a "Hold" control statement was issued immediately prior to the issue of the "Call" statement.

One final routine to be discussed is the low-core cell #22 entry routine, "Chain". Its purpose is to call a named program from a designated library medium, then execute it without packing any arguments. This provides a method of segmenting a program which is too large to fit into the available core memory along with its "Common Storage Data Arrays". Thus the segments can be called into core one at a time for execution. Each segment over-writes the previous segment. Processed data from one segment to the next is inter-changed via designated common storage. This idea is extended one step further in the COOP MONITOR system by providing both segmenting and over-lay modes of operation.

The remaining resident routines, which were added to resident by the authors, shall be discussed in section 4.2 and section 4.4. These sections concern the "Data Display Programming" and the "Satellite Station Programming".

2.4 System Characteristics

Perhaps the most significant characteristic of the Fortran 60 System is that in its original form, its Resident control programming including all equipment control, job sequencing monitor, and basic service routines occupied only 4000 octal cells of core storage. The next most important feature of the system is its simplicity of organiztion when compared with the COOP MONITOR system. This is due, of course, to its limited scope. For some uses this simplicity is of great importance. Later the COOP MONITOR system will be discussed, at that time we shall see

that it has many advantages over the simplier Fortran 60 System. With this comparison to make, it is necessary to list the relative advantages and disadvantages of the Fortran 60 System.

2.4.1 Advantages

a. SIZE; As mentioned, the basic control programming was a modest 4000 octal cells The authors' additional programming to provide a remote satellite computing station, and cathode ray tube data display facility increased the resident size to only 5000 octal cells. This remains a fairly acceptable figure.

b. FAST STACKED JOB PROCESS TIME; The Fortran 60 monitor operates in a mode which allows re-use of the Fortran Compiler by many different jobs. This saves library reference time and hence saves process time, provided the compiler remains undamaged. The compiler compiles in memory only and uses no scratch tapes, this also leds to a very fast compilation. There are disadvantages to both these techniques which over shadow the advantages. These will be discussed as listed disadvantages.

c. MINIMUM INPUT / OUTPUT UNITS REQUIRED; It is possible to operate the Fortran 60 System with only three I/O units operating. This can be a big advantage when several I/O units are inoperative or are assigned for utilization by the center's 160 computers.

d. OPERATOR SERVICE ROUTINES; A well developed repertoire of operator service routines are an important advantage to a system. Fortran 60 has several such routines such as; endfile mark tapes, rewind, lock, duplicate tape, verify, list, etc. Given a little time, the COOP MONITOR System will probable incorporate several of these routines.

e. OPERATOR TRAINING TIME; At the Postgraduate School many students are given instruction, check-out, and direct access to the school's computers. As many as 250 students may participate in these laboratory periods per year. Fortran 60, with its simplicity provides a good vehicle for instructional purposes.

f. MONITOR / OPERATOR REQUIREMENTS; The Fortran 60 monitor system requires a minimum of operator intervention to process a stacked job run. Unless one of the jobs "blows the system Off-Line", ie ; manages to execute a storage or jump instruction erroneously, the system requires no operator assists. The COOP MONITOR requires considerable operator manipulation of "Jump Keys" and loading / unloading of tapes, etc. Thus the Fortran 60 System normally requires less operator supervision.

g. LIBRARY MAINTENANCE; Library maintenance and updating should not be a daily occurance, however it is required frequently. Therefore a quick and reliable system to accomplish this task is required. Fortran 60 has three routines and a simple organization which make this an easy job. The routines are; transfer, vfylib, and list. In contrast, the COOP MONITOR System's "Libedit" routine is rather complex, is indirect via punched cards, and requires apriori knowledge concerning the current "directory" of routines on the library. It would be nice, if a directory dump routine was written and also an individual routine verify capability is required.

h. MIXED FORTRAN and SYMBOLIC STATEMENTS; Fortran 60 incorporates a fortran compatible symbolic language. Programs written for this system can inter-mix both symbolic and fortran statements on a line for line bases, if desired. For many uses such as programming shifts, logical and masking operations, searches, and other computer operations which the Fortran 60 language doesn't provide, this ability has a distinct advantage.

i. INTERRUPT ROUTINE USAGE; The only usage of the interrupt routine in the Fortran 60 System is for providing a real time clock. This would appear to be a gross neglect of such a powerful equipment capability. The advantage, here, is that the provision for a satellite station operation requires much usage of this interrupt routine, hence having a system which doesn't utilize the interrupt in some complex manner is unique.

2.4.2 Disadvantages

a. COMPILER VULNERABILITY; As mentioned under advantages, the Fortran 60 compiler is normally called into the computer's memory only when initializing monitor processing of a stacked job run. Thus there exists the possibility that one of the individual jobs may possibly damage the compiler. There after the remaining jobs of the stack are forced to be compiled by a compiler which is no longer reliable. This can result in significant loss of computer and programmer's time, since the programmer is led to believe his program was incorrect.

b. SLOW TAPE HANDLING TECHNIQUES; Two standard length tape records are employed. Standard binary records are 54 words long. Standard BCD records are 15 words long, thus providing for 120 character records. This standardization is an advantage in some instances. However, the disadvantage is the cost in time which must be paid since the tape motion must be stopped and started between each record. The COOP MONITOR System solved this problem quite well. A second detriment to tape handling is the lack of utilizing the buffer input / output channels to advantage. The only usage of the buffer capability is employed under monitor process control. In this instance, very poor usage is made since there is a complete disregard for error checking during input / output operations. This slow tape handling characteristic is a serious

one since much time is spent by the system in searching the library tape. The library tape can grow to be of considerable length due to the considerations discussed as a disadvantage due to "Multiplicity of Subroutines on the Library". COOP MONITOR solved this problem by providing long variable length library records and a directory which prevents searching the full length of the library tape for routines which are "not available".

c. PROGRAM SIZE LIMITATIONS; Since Fortran 60 doesn't utilize scratch tapes during compilation, it requires that memory must hold the resident program (4000) plus the compiler (14000) plus the compiled version of the user's program plus an undefined number of cells outside the compiler which are used by the compiler for tables and temporary storage. Thus the programmer losses at least 20,000 octal cells. This is a serious limitation in many large programs. COOP MONITOR has improved this situation, but at a considerable cost in speed, due to the necessity of reloading the compiler for each job and also having to provide an intermediate and "Load and Go" tape.

d. INEFFICIENT SPACE COMPILATION; Fortran 60 compiles the full size of arrays and reserved space into the binary object program (ie. card images corresponding to the compiled program) at compile time. This coupled with the compilation limitation mentioned previously tends to further hinder efficient usage of memory space during compilation of programs. One advantage this does provide is that arrays and reserved spaces are initially cleared in a Fortran 60 program. This is not true of a COOP MONITOR compiled program.

e. MULTIPICITY of SUBROUTINES on the LIBRARY; Fortran 60 requires that all subroutines called by a program be available at compile time. The reason for this is that they are embedded into the compiled program as it is compiled. If this compiled program is then added to the library, the library is unnecessarily lengthened. This is due to the fact that versions of the existing subroutines are embedded in many programs. Not only does this scheme of compilation lengthen the library tape and thereby slow the system down, but it also creates very serious library maintenance problems. Thus the simple corrections or updating of a subroutine isn't so simple anymore. In order to correct a subroutnie, it is necessary to re-compile all programs which use the subroutine. This is due to the embedding effect. This can get to be a big bookkeeping problem and can create a lot of work. Again the COOP MONITOR System has solved this problem.

f. NO PROVISIONS FOR MIXED BINARY / BCD INPUT PROGRAMS; Fortran 60 does not provide any method for inputting a binary program or subroutine into a stacked monitor run, if it is not already a part of the library. Thus no mixed input to the system is allowed. This requires frequent recompilation of subroutines and programs which are not incorporated onto the library tape. Again the COOP MONITOR / FORTRAN 62 System provides the solution.

g. LIMITED "DE-BUGGING" AIDS; The only automatic de-bugging aids incorporated into the Fortran 60 System are the compiler and arithmetic error print-outs. The selectable aids consist of a "Map Language" (see ref. #21, pg #102) listing of the compiled program, and a few dump routines which were never very effective. A magnitude improvement has been made by the COOP MONITOR System towards providing de-bugging facilities, both automatic and optional.

h. UNDESIRABLE INTERMEDIATE LANGUAGE FORMAT; The intermediate language of the Fortran 60 compiler is "Map Language". This form of computer coding can become familiar, but only a small percentage of the general users of the computer facility were ever motivated to use it. Three "languages" for coding a single compiler seems to be more than should be required. Successful utilization of the Fortran 60 System requires a knowledge of the fortran language, CDC Symbolic format, and of the intermediate language, "Map". The Fortran 62 system is a two language system and is quite acceptable except for the exclusion of the line for line inter-mix capability between fortran and symbolic coding.

i. POOR INPUT / OUTPUT FLEXIBILITY UNDER MONITOR CONTROL; The Fortran 60 System originally had a very weak tape assignment system. It would assign tapes for scratch when the units weren't even loaded. This was corrected by the authors for the school's facility. Another deficiency is the inability of the monitor to provide multiple input / output BCD units. This is desired in many programs, for instance, when cumulative data is being generated and processed over a long period, or when special table lists are to be formed by a program, etc.

j. SLOW CARD READ / PUNCH ABILITY; The school's "On-Line" card facility is a CDC 1609 system which is capable of operating at either 50 or 100 cards per minute. The Fortran 60 system provided operation only at the 50 card per minute rate. COOP MONITOR will run at the maximum speed of 100 cards per minute and compiles as it reads if the card reader is being used for input. This rate is still too slow for a general usage on-line input medium, hence it is used only as a back-up system for the IBM 1401 System which normally processes the card to tape conversion "off-line".

3.0 COOP MONITOR SYSTEM

The COOP MONITOR SYSTEM was developed to allow a computer facility the ability to operate with only one "library" tape and yet effectively process any type of input programming or source language. This would eliminate the need to sort incoming jobs into separate groups to be run under individual library tapes. The advantage of a single library is readily apparent when operation from a remote station is anticipated, where the operator is physically separated from the main computer and unable to change the current system library tape.

To increase COOP MONITOR'S flexability it is organized into two basic levels of control. At the highest level is the Master Control System (MCS), which maintains the overall system coordination and supervision. At the next level is the Secondary Control System (SCS), and there may be any number of these available on the library. Each SCS is designed to perform a particular job or combination of jobs depending on the requirements of the computer facility.

A brief description will now be given covering the basic organization and operation of COOP MONITOR, for a more detailed description refer to ref.10.

3.1 Master Control System

To accomplish the desired system supervision, MCS is organized into nine functional subroutines and the Bootstrap loader (BOOT), see figure 3. BOOT and the first eight subroutines make up the "permanent" resident, with the memory space occupied by the Job Sequencer (6000 cells) released for use each time control is given to the Secondary Control System. After the SCS has finished using the subroutine

"Loader" the memory space it occupied may also be released for use by the current program. This procedure allows MCS to be very large and complex (13500 cells) and yet not unduly penalize an input program in memory space requirements. The basic functions of each subroutine are as follows:

a. BOOT; Contains the Available Equipment Table (AET), which enables MCS to determine the input/output units available to the computer and their intended use. This routine will load on option either MCS or the system recovery dump routine "EXTREC". BOOT also contains the MCS flags to store current job information to allow intra-job continuity.

b. CLOCK; Allows use of the computer real time clock to set a time limit on the current job. This routine is available to the programmer to set one additional time limit and also may be used to determine elapsed time.

c. MEMREC; Maintains a record of the limits on the one contiguous block of available memory between resident and any programs loaded into high core.

d. OPCON; All communication between the computer operator and MCS or SCS is thru this routine. This input or output is normally via the console typewriter.

e. INTERRUPT; This routine has three sections, SELECT, DETECT, and REMOVE. These allow for the selection and detection, or the release of any of the internal computer interrupts or on channels 1 to 6 inactive.

f. READ/WRITE; This subroutine controls all input/output operations of the computer. Provides protection from illegal use of any unit and prevents interference between units on the same channel.

g. ERROR; Used in the event of user program failure or the detection of illegal operation by MCS, entry into this subroutine will terminate the current job. ERROR will output the current contents of all the console registers (on the standard output medium) and the lower address of cell 7, and then determine if the user desires a memory dump.

h. RW1607; The driver for the magnetic tape units and the only driver contained within MCS. This subroutine is the only place where tape units are selected and activated, except in BOOT for loading MCS or EXTREC.

i. LOADER; Loads all relocatable binary programs or subroutines either from the system library or user provided input medium. This subroutine packs programs into high core memory with common block storage assigned to low core just following "LOADER".

j. JOB SEQUENCER; Used between input jobs to set program limits, determine and load input/output drivers requested by the user, and to call and transfer control to the desired Secondary Control System (SCS). This subroutine also contains the programming to process any computer operator statements received via "OPCON".

These functional blocks are closed subroutines with multiple entry points depending on the particular function desired. These entry points are available to all routines and must be used whenever one of the above operations is necessary. This also includes other blocks within MCS, example, The JOB SEQUENCER

BLOCK DIAGRAM COOP MONITOR

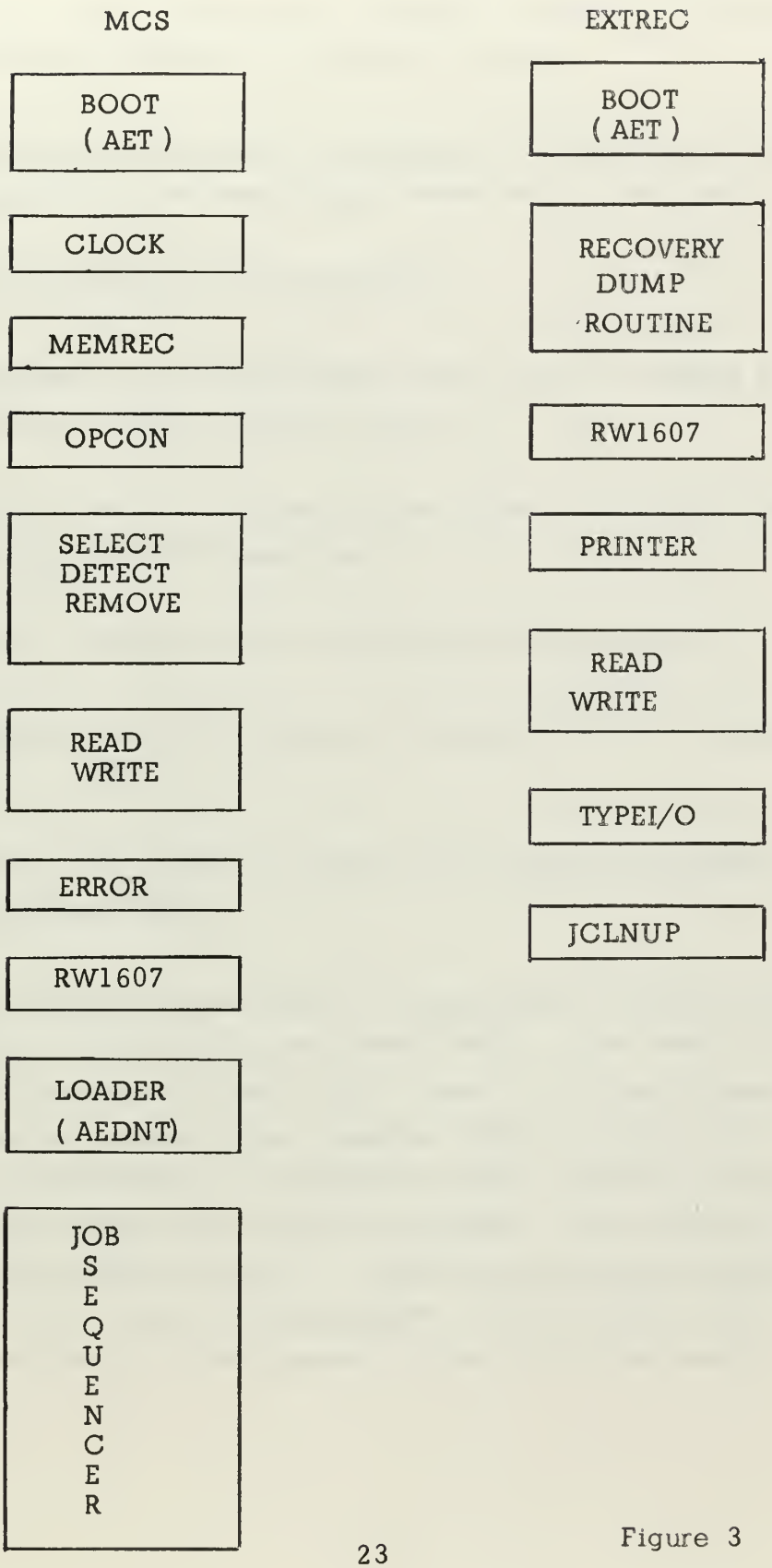


Figure 3

must use LOADER to make a library call, and LOADER in turn must use RW1607 to actuate and read the magnetic tape unit. These subroutines operating together as a Master Control System provide the following features:

a. Automatic, variable job sequencing, including accounting records and time and output line limits set for each job.

b. Operator-Machine communication.

c. Assignment of all input/output units and the loading of the necessary drivers for these units.

d. Single linkage to all input/output drivers to protect them from interference and illegal use.

e. Flexible linkage to the CDC 1604 interrupt features.

f. Automatic recovery features in the event of "user" program failure.

g. Calling and tranfer of control to any desired Secondary Control System (SCS).

The assignment and control of input/output equipment by the MCS is accomplished thru three tables. The first, the Available Equipment Table (AET), lists all the input/output equipment connected to the computer. Some of these I/O units are designated as "standard" (SIO), and are assumed to be always available for use by the system. The entries in the AET (one for each I/O unit), contain the following information: Allowed use, current job assignment, standard unit or not, equipment type, and the location of the I/O driver name. For

a complete description of the AET entry format see ref.10. This table is set for each installation and is changed only as equipment status changes. A listing of the current AET is shown in figure 6c.

The second, the Available Equipment Driver Name Table (AEDNT), lists the name and entry point location for each I/O driver subroutine for controlling the units listed in the AET. The entries are two words long. The first word contains the entry point locations of the routine if it is in memory, or an exit to MCS if not in memory, see section 3.5 part D for an example of an AEDNT entry.

The third, the Running Hardware Table (RHT), lists the input/output assignments of units for the current job only. The entries are AET entries with the address of the proper driver entry point inserted and the allowed use set. Logical unit numbers are used as an index to this table, logical numbers 1-49 are available to the programmer, 0 is reserved for the system library tape and 50-63 for standard units.

3.2 Secondary Control Systems

The great flexibility possible with the COOP MONITOR is in the use of a Secondary Control System (SCS) to control the processing of a specific job. There can be any number of Secondary Conrol Systems which may be written by each computer facility to handle the specialized needs of its users. Thus for use at the Postgraduate School the standard "stacked job" daily production runs would be controlled by one SCS, and a satellite digital controller simulation by a different SCS, and a real time radar intercept problem by a third SCS. All of these SCS would be placed on the standard library tape and available within a few seconds after a particular one is called from the library. No modification is needed in the standard COOP MONITOR

to use a different SCS. The name of the desired SCS is given to the JOB SEQUENCER in MCS at the begining of each job and this name will go to LOADER and the SCS will be loaded from the system library.

There are two SCS provided in the present version of COOP MONITOR, these are called "COOP" and "LIBEDIT". A brief description of each is given to illustrate the technique of tailoring a SCS to do a specific problem. The first, COOP, is responsible for the intra-job sequencing necessary to compile and execute an input job, ie; the calling of compilers or assemblers, transferring binary card images to a "load and go" tape, calling LOADER to load the "load and go" tape and finally, execute the loaded program. The present version of COOP can call two compilers, FORTRAN-62 and CODAP1 (actually an assembler), but COOP can be modified to accept any number of compiler languages. COOP also contains internal routines for user controled binary deck manipulation, loading and execution.

The second, LIBEDIT, has two subroutines, Edit Library Tape (ELT) and Prepare Library Tape (PLT). Edit Library Tape is used for minor changes in the system library such as a deletion or addition to the subroutine file or within MCS and EXTREC. This secondary control system makes a copy of the present library tape upto the point of the desired change and then inserts or deletes the proper information. For a detailed description of the steps necessary in a complete LIBEDIT, see appendix I and ref.17. Prepare Library Tape is used for major changes to the system library. The user must now supply all the information necessary to produce a new version of the system library tape, nothing is copied from the old library tape. LIBEDIT does not provide any means of verifying the portions of the new library tape that were not changed.

3.3 System Operation

A brief description of the operation of MCS can now be given, for a more detailed description see ref.10. The first record of the library tape, BOOT, see figure 4, is read into the computer by manually selecting and activating the proper magnetic tape unit. BOOT is then executed at cell #2, see figure 6, and will verify the AET table and then skip over the next record on the library tape, EXTREC, and load MCS, see figure 4. Control is now given to MCS which determines the standard input/output units (SIO), loads the necessary drivers for those units if they are not in memory, fills the unused memory space with a jump to ERROR instruction to trap user program failures, and then notifies the computer operator the system is ready to accept job assignments. The operator then either signals MCS to alter the AET or SIO assignments or to begin processing the "stacked job" on the standard input. When signalled to begin processing, MCS reads the control card from the standard input, logs the first job on the standard outputs (list and punch tapes and the console typewriter), and sets the time and output line limits for the current job, the recovery key is stored in BOOT. The input/output units needed, in addition to the standard units, by the incoming job are determined and assigned and if the proper drivers are not already in memory they are called and loaded. MCS then releases the memory space occupied by the non-permanent section of MCS (Job Sequencer) for use by the SCS or the input program. The desired secondary control system is determined and loaded into memory and MCS then transfers control to this SCS, see figure 5. When the SCS has finished processing the current job, control is returned to BOOT with an indication for a "normal" or "abnormal" job termination. For a "normal" termination BOOT reloads MCS

COOP MONITOR LIBRARY TAPE FORMAT

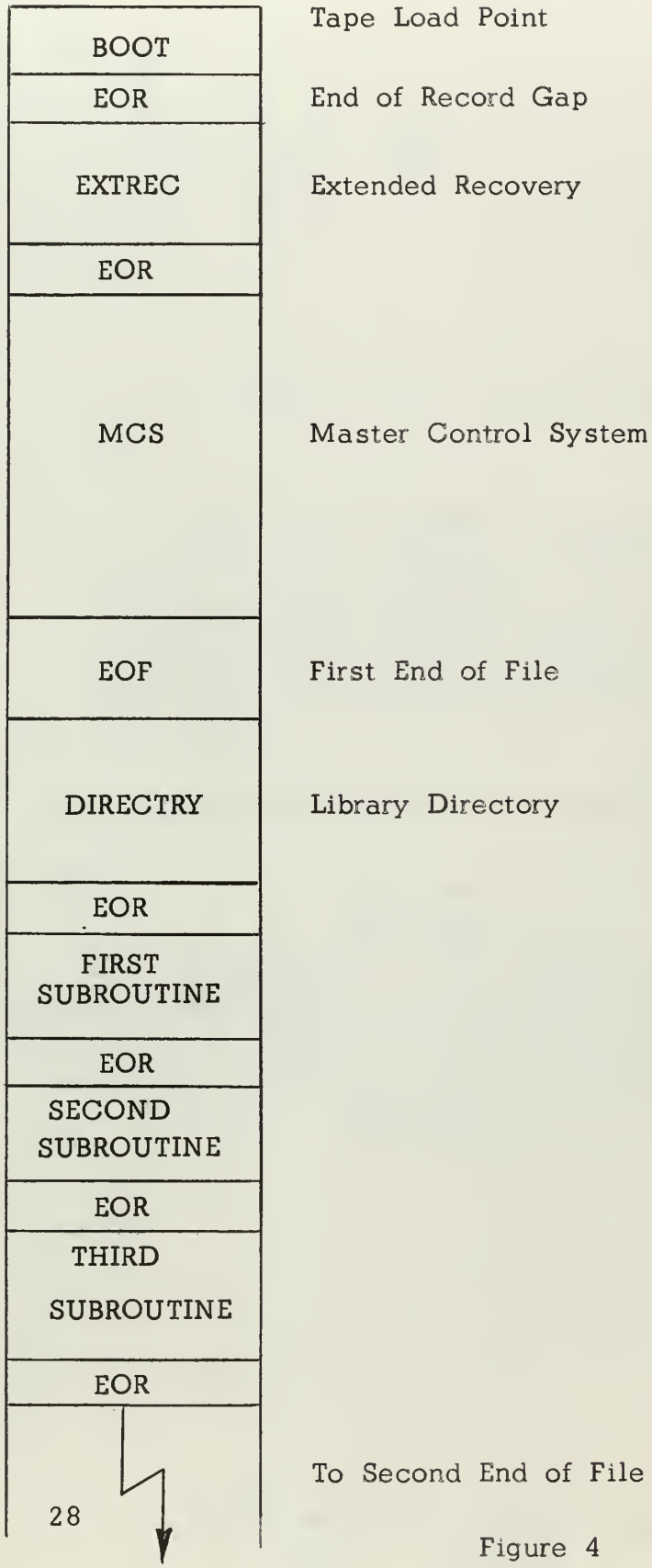
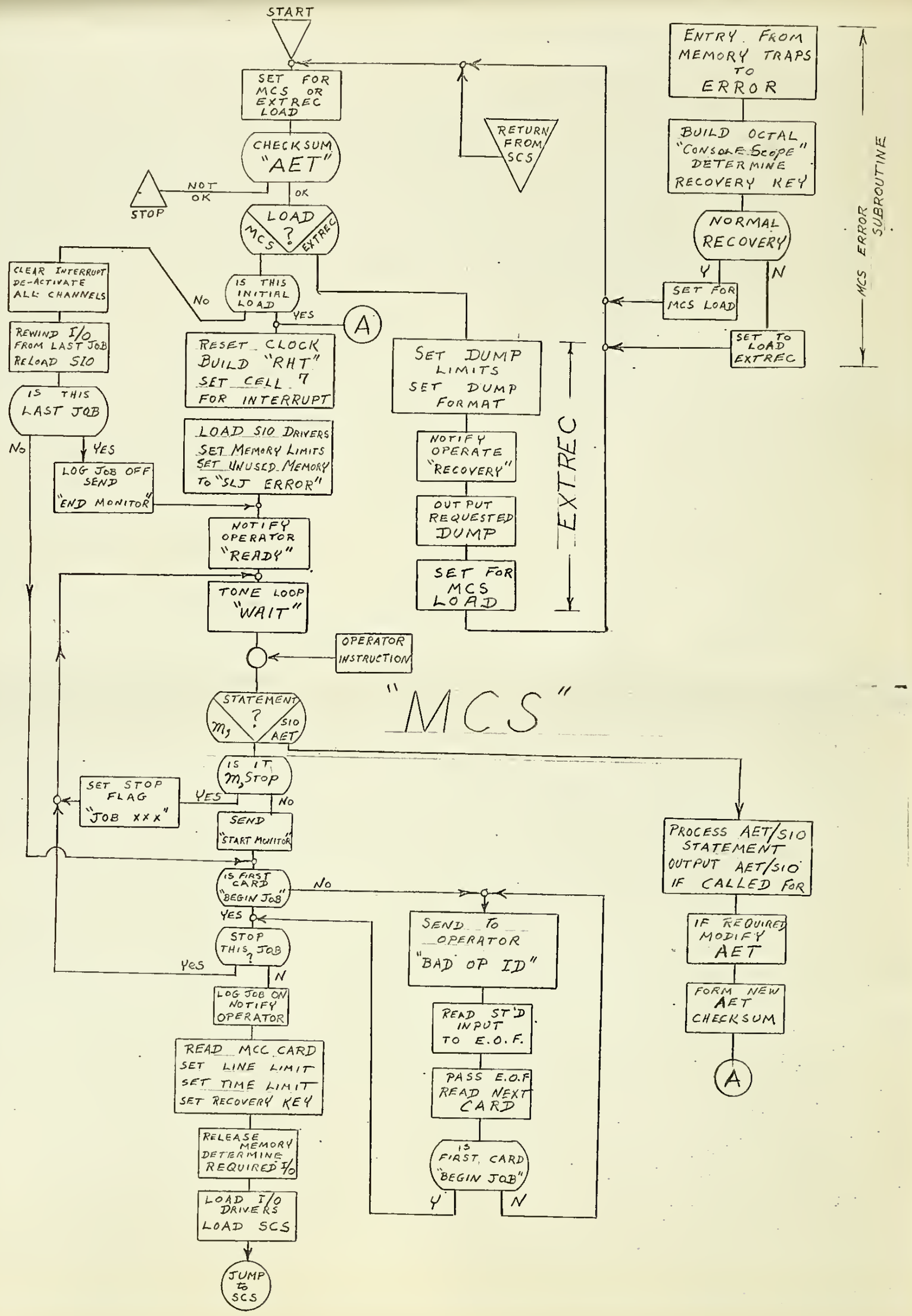


Figure 4



COOP MONITOR SYSTEM MCS

FLOW DIAGRAM

Figure 5

and if there are more jobs on the standard input, the cycle is repeated. For a "abnormal" (error) termination a "Console Scoop" (the contents of all the console registers) is prepared and sent to the standard output tape to aid the programmer in debugging efforts. The recovery key, set by the user at the start of the job, is examined to determine of a memory dump is desired, if not control is given back to BOOT for a MCS reload and the process continues as before. When a memory dump is desired control is returned to BOOT for a EXTREC load, see figure 6. Extended recovery then provides the desired memory dump on the standard output tape and returns to BOOT for a MCS reload. Again if there are more jobs on the standard input tape MCS will start the sequence over, otherwise it will terminate this run and notify the computer operator.

3.4 System's Characteristics

Once over the initial shock of working on a system of this size, the next effort was to consider COOP MONITOR'S good and bad characteristics when the special requirements of the Postgraduate School computer center were included in the system. These special requirements included the daily "stacked job" production runs of short duration, longer runs by the students during the evening, the increased use of the satellite capability of the schools computers, and the addition of a sophisticated digital display unit (Data Display Inc. DD65) as a remote computer input/output device. In this light the COOP MONITOR was determined to have the following major advantages and disadvantages:

3.4.1 Avantages

1. The system can handle stacked input jobs written for multiple compiler languages. The present version includes Fortran-62 and Codapl and the addition of the Neliac compiler and the Scrap assembler is possible.

2. Operates with multiple secondary control systems each specifically designed for a job. Present system has two, COOP and LIBEDIT and others may be added to process satellite and remote display problems.

3. Automatic, programmer controlled, debugging aids such as the "console scoop", provided in the event of program failure and the "snap" feature for those that do not fail but also do not produce correct results.

4. Automatic system recovery in the event of failure by the input program.

5. All input/output is thru one routine, therefore it is very difficult for an input program to illegally use any I/O unit. Interference between units is also minimized.

6. COOP can accept mixed, binary and source language, input programs.

7. Magnetic tape driver routine uses variable block read and write for maximum speed when controlling the system library tape. All read/write is buffered, with parity and length checking.

8. Variable standard input/output units are allowed and these can be changed before each job.

9. Multiple input/output units for each job are allowed up to the maximum available to the computer.

10. The system is capable of handling very large input programs as MCS releases memory and the compiler used is over written when the program is loaded for execution. For programs that are still too large, segmenting and overlaying are provided.

11. The compiler used, Fortran-62, has special features all its own; easy to read intermediate language (Codap1), buffer operation for input/output, and single library subroutine call for each program.

12. Compiler is reloaded for each job thus assuring a clean start.

13. The system will accept Fortran-63 when it is available.

3.4.2 Disadvantages

1. The system is very large (13500 cells) and complex, making it difficult to gain the understanding necessary to perform any system modifications.

2. Many complex (at first) control cards are necessary to process each input job.

3. Provides an unused and unsuppressable accounting information output on the paper tape punch.

4. Normal system operation requires a large number of standard input/output units to be operating. Four magnetic tape units (library, punch, input, and output), the paper tape punch for system accounting, and the console typewriter are required to satisfy the "standard" I/O requirements.

5. Inability of the computer operator to add or delete a standard input/output unit (the type of I/O unit can be changed).

6. Step by step operation by the computer operator via the console typewriter is not possible.

7. The interrupt schedule is not compatable with a real time problem solution in a micro-second time sharing mode, due to the frequent use of interrupt lockout. No provision is made for operation from a remote (satellite) station.

8. No programmed restart available for use by non-autoload CDC 1604 computers (the autoload must be simulated by manually performing the required instructions).

9. No operator service routines are available such as call, copy, transfer, verify, etc.

10. Program errors are sent only to the standard output tape requiring the user to list this tape before determining any errors.

11. The system would not output/input on the CDC 1609 card punch/read unit available at the computer facility.

3.5 Alterations To Coop Monitor System

After a basic understanding of the Coop Monitor System was obtained, it became apparent some modifications would be required before the system would be acceptable for operation at the Postgraduate School's computer facility.

The first modification attempted was the alteration and corrections needed in the cardread and cardpunch drivers to allow operation with the CDC 1609 card read/punch unit. In addition to the corrections, it was also necessary to rewire the unit's plugboard to make the read/punch format compatible with the Coop Monitor. These read/punch routines are an improvement in speed to 100 cards per minute vice the 50 cards per minute available with Fortran-60. Once the drivers were corrected, they were recompiled and the binary versions

were added to the system library tape using the secondary control system "LIBEDIT". For a complete description of the LIBEDIT procedure, and the input job format see appendix I. The correction of these subroutines provided a library tape more suited to the needs of the school computer center but still left some major deficiencies in the system that would prevent convenient operation. These were the following:

a. The unsuppresable accounting medium being output on the paper tape punch, requiring the punch to be selected at all times. This made the paper punch unavailable to the user as an output device.

b. No provision for a programmed restart for operation on a non-autoload computer (CDC 1604). In the event of MCS failure a manual "bootstrap" is required.

c. Error messages from MCS "LOADER" are not available to the user except by listing the standard output medium.

d. No provision for handling operation from a remote station.

The correction of these deficiencies all had one thing in common, they would all involve changes within the structure of MCS. Many early failures soon made it apparent changes in MCS could not be made without a complete understanding of the internal organization of MCS, and a positive knowledge of the operation of LIBEDIT. After these two programs were more thoroughly understood, the modifications resulted in the successful production of a new library tape providing these new features:

a. Accounting medium suppressed, the paper tape punch is not selected unless the operator intends to use it as an output medium.

b. A programmed restart is available, at cell #30, and this will simulate the "autoload" feature of the CDC 1604A except for the initial load.

c. All error messages from MCS "LOADER" are now also output on the comment to the operator medium, normally the console typewriter.

d. All linkages necessary (AET and AEDNT entries) for the calling and execution of a driver subroutine for the remote display (DD65) are now in MCS. Space required for this driver is reserved (1000 cells) within the "permanent" section of MCS just prior to LOADER.

These modifications now provide a library tape that should be much more convenient for the computer center to use and at the same time provide a system that will readily accept experimental drivers and subroutines for operation from the remote station. The detailed explanations and coding involved in these modifications are shown in the following sub-sections.

A. Modifications to BOOT

The postgraduate schools computer (CDC 1604) is a non autoload machine and thus requires a manual selection and activation to read the library tape. The COOP MONITOR uses the autoload feature anytime there is a failure of the automatic recovery routine. To avoid the necessity of this manual "bootstrap" each time this may occur a simulated autoload was placed into BOOT starting at cell #30. This allows an "autoload" restart when needed. At the same time the standard input/output assignments in the Available Equipment Table (AET) were altered to provide the following:

a. Library on channel 3/4, tape unit #1
b. Input on channel 3/4, tape unit #2
c. Available for use, channel 3/4, tape units #3 and #4
d. Output on channel 5/6, tape unit #1
e. Punch on channel 5/6, tape unit #2
f. Scratch 1 and 2 on channel 5/6, tape units #3 and #4

```
START MONITOR RUN.
BEGIN JOB 231
COOP,,,S/2S,20,50000.
CODAP1,1,1,.
1                                                IDENT     BOOT
        RANGE                    FWA   -   LWA+1
                               00002       14557

        ENTRY POINTS
                               01205        READ*
                               01212       WRITE*
                               00655      SELECT*
                               01140      REMOVE*
                               00743      DETECT*
                               01763       GETCH*
                               01770      CHKSTD*
                               04063      LOADER*
                               04064     RELOCOM*
                               02740      LIBREW*
                               02657     CLBNBCD*
                               02616        EXIT*
                               01777       ERROR*
                               01174     MODIRET*
                               00401       BCDBN*
                               00335       OPCOM*
                               02233      RECRET*
                               02252      RECLIM*
                               00524     FLAGTST*
                               00217      MEMREC*
                               01742         CBF*
                               01743        CBF1*
                               01744        CBF2*
                               01751        CBF7*
                               00714     IOSLECT*
                               01473      MRW600*
                               06342        LOK1*
                               05745      LMSRCH*
                               01121         ILMF*
                               01752          FCH*
                               01034       EXSEN*
                               00167      SETCLK*
                               00076        DATE*
                               00061        SIEOF
                               02732     LIBPOSIT
        NO EXTERNAL SYMBOLS
                               00002               ORG     2
          00002     50   1     00045               ENI     1  CHECKSUM-9          SET X1
                    13   0     00057               LAC        CHECKSUM
          00003     50   0     00003               NOP        3
                    74   3     40000               EXF     3  40000B              TERMINATE BUFFER
```

```
00004   14  1  00010                ADD   1  8                 ACCUMULATE
        55  1  00004                IJP   1  *                  CHECKSUM
00005   22  0  00006                AJP   Z  *+1
        75  0  00014                SLJ      CKSUMERR
00006   20  0  00003                STA      3
        74  0  00070                EXF      70B               CLEAR ARITHMETIC FAULTS
00007   75  1  00023                SLJ   1  LOAD              JUMP IF SW 1 IS ON (LOAD 2ND
        75  0  00022                SLJ      PASSONE
00010   50  0  00000  DISASTR*      NOP
        74  0  00101                EXF      101B              REMOVE INT. ON ARITH. FAULT
00011   50  1  00046                ENI   1  CHECKSUM-9        SET X1
        13  0  00057                LAC      CHECKSUM
1 00012 14  1  00010                ADD   1  8                 ACCUMULATE
        55  1  00012                IJP   1  *                  CHECKSUM
00013   74  0  00070                EXF   0  70B               CLEAR ARITHMETIC FAULTS
        74  0  00100                EXF   0  100B              SELECT INTERRUPT ON ARITH. F
00014   22  0  00015  CKSUMERR      AJP   Z  LOADON
        76  0  00014                SLS      *                 CHECKSUM ERROR STOP
00015   74  0  32011  LOADON       EXF   0  32011B            SELECT READ TAPE 1
        74  0  32005                EXF   0  32005B            REW 1
00016   57  0  00003                SIL   0  3                 SET TERMINAL ADDRESS TO ZERO
        74  7  32000  -             EXF   7  32000B            WAIT
00017   23  1  00022                QJP   N  PASSONE
        75  0  00021                SLJ      *+2
00020   50  0  00000                NOP                        TEXAS BOOT
        75  4  01777                RTJ      ERROR*            TEXAS BOOT
00021   74  3  77777                EXF   3  -0                PASS ONE RECORD
        74  7  32000  -             EXF   7  32000B            WAIT
00022   74  3  77777  PASSONE      EXF   3  -0                PASS ONE RECORD
        74  7  32000  -             EXF   7  32000B            WAIT
00023   74  3  00077  LOAD         EXF   3  EXECUTE           LOAD NEXT RECORD
        74  7  32000  -             EXF   7  32000B            WAIT
00024   74  7  32003                EXF   7  32003B            EXIT ON NO PARITY
        75  0  00026                SLJ      *+2
00025   74  3  00000                EXF   3  0                 TERMINATE BUFFER
        75  0  00077                SLJ      EXECUTE           JUMP TO LOADED PROGRAM
00026   74  3  00000                EXF   3  0                 TERMINATE BUFFER
        76  0  00077                SLS      EXECUTE           STOP ON PARITY, OPTION TO CO
00027   50  0  00000                NOP                        SPARE CELL
        50  0  00000                NOP
00030   74  0  32015                EXF      32015B            AUTO LOAD ENTRY
        74  7  32000  -             EXF   7  32000B            WAIT
00031   74  3  00002                EXF   3  2                 READ IN BOOT
        74  7  32000  -             EXF   7  32000B            WAIT
00032   75  0  00002                SLJ      2                 JUMP TO BOOT
        50  0  00000                NOP
00033   77  7  77777  AET          OCT      -0
        77  7  77777
00034   31  0  13201                31    0  13201B            SYSTEM LIBRARY
```

```
                04   0   06440                    04        AEDNT+1
00035           31   0   23202                    31    0   23202B          STANDARD INPUT
                04   0   06440                    04        AEDNT+1
00036           30   0   53203                    30    0   53203B          AVAILABLE
                04   0   06440                    04        AEDNT+1
00037           30   0   63204                    30    0   63204B          AVAILABLE
                04   0   06440                    04        AEDNT+1
00040           31   1   55201                    31    1   55201B          STANDARD OUTPUT
                04   0   06440                    04        AEDNT+1
00041           31   1   65202                    31    1   65202B          STANDARD PUNCH
                04   0   06440                    04        AEDNT+1
00042           30   2   15203                    30    2   15203B          SCRATCH 1
                04   0   06440                    04        AEDNT+1
00043           30   0   75204                    30    0   75204B          SCRATCH 2
                04   0   06440                    04        AEDNT+1
00044           10   1   01410                    10    1   01410B          1609 PRIMARY READ
                01   0   06442                    01        AEDNT+3
00045           10   0   01420                    10    0   01420B                SECONDARY READ
                01   0   06442                    01        AEDNT+3
1       00046   20   0   01402                    20    0   01402B          1610-1-PUNCH
                01   0   06444                    01        AEDNT+5
00047           30   0   07100                    30    0   07100B          INPUT FROM DISPLAY
                06   0   06460                    06        AEDNT+17
00050           11   0   01110                    11    0   01110B          TYPEWRITER INPUT COMMENT U
                00   0   06450                    00        AEDNT+9
00051           21   0   01110                    21    0   01110B          TYPEWRITER OUTPUT COMMENT
                00   0   06450                    00        AEDNT+9
00052           20   0   01122                    20    0   01122B          PTP
                03   0   06452                    03        AEDNT+11
00053           10   0   01121                    10    0   01121B          PTR
                03   0   06454                    03        AEDNT+13
00054           21   0   01124                    21    0   01124B          PTP - ACCOUNTING
                03   0   06456                    03        AEDNT+15
00055           30   0   07000                    30    0   07000B          OUTPUT TO DISPLAY
                06   0   06460                    06        AEDNT+17
00056           C0   0   00000  AETEND   OCT          0
                00   0   00000
00057                            CHECKSUM BSS          1
00060           00   0   00000  JSFLAG   OCT          0
                00   0   00000
00061           00   0   00000  SIEOF    OCT          0
                C0   0   00000
00062           00   0   00000  LINECNT  OCT          0
                00   0   00000
00063           00   0   00000  JOBXXX   OCT          77
                00   0   00077
00064           00   0   00000  JSTOPX   OCT          0
                C0   0   00000
00065           00   0   00000  RDATA    OCT          0
```

```
          00   0   00000
00066     00   0   00000   RPROG     OCT       0
          00   0   00000
00067     00   0   00000   RDMPFLG   OCT       0
          00   0   00000
00070     00   0   00000   MARKTIME  OCT       0
          00   0   00000
00071                      LASTOT    BSS       1
00072                      OLDTIME   BSS       1
00073     37   7   77777   TOTIME    OCT       3777777777777777
          77   7   77777
00074     00   0   00000    RHT      OCT       0
          00   0   00000
00075     00   0   00000   REMTIME   OCT       0                          IN BOOT
          00   0   00000
00076     00   0   00000   LASTCELL  OCT       0
          00   0   00000
                   00075   DATE*     EQU       LASTCELL
                   00077   EXECUTE   EQU       LASTCELL+1
```

All references to channel 1/2 magnetic tape units were removed (the school has no units available) and the I/O linkage for the remote display driver was inserted, see figure 6.

These modifications to BOOT altered the location of the begining of the AET and the subroutine EXTREC uses this location as an absolute address. This required recompiling EXTREC with the correct address for the AET and then placing this on the system library tape again using the routine LIBEDIT.

B. Suppression of Accounting Medium

The original version of COOP MONITOR provided for accounting information to be output on the paper tape punch. This output was of unknown format and therefore of no use to the computing center. This did demand the paper punch be selected and prevented the use of the punch as an output device. The suppression of accounting output was accomplished by taking advantage of the READ/WRITE subroutine in MCS. All I/O requests must be thru this routine and this includes any requests for output to the accounting medium, normally the paper punch. Logical number 55 is reserved for the accounting medium and to suppress the output a test for logical 55 is made each time READ/WRITE is entered. If the entry is for any logical number except 55 the request is processed. If the request is for logical 55 a no action exit is made.

This test is made by adding the following programming to the READ/WRITE section of MCS at approximately cell 1225.

```
INA      -55        Test for accounting
AJP  N   *+3        Jump if not accounting
INA      2          Build exit to suppress
SAU      MRW800        accounting output
SLJ      MRW800     Jump to exit
```

This suppresses the accounting output but still allows the paper tape punch to be used as an output medium if desired. Unless the punch is to be used as an additional output unit is need not be selected.

C. Loader Errors to the Operator

Many of the failures of programs to run under the COOP MONITOR result from errors detected by the subroutine LOADER in MCS. These errors cause messages to be written on the standard output medium and this must be listed off-line (unless the standard output is an on-line printer) before the operator can determine the cause of the error. To allow the error message to also be output on the comment-to-operator medium, normally to the typewriter, the following programming was added to the subroutine LOADER in MCS at approximately cell 6274.

```
RTJ     WRITE     Errors to operator
01      53        BCD              Unit #
03      LERR      Write/Check      FWA
00      LERR+5                     LWA+1
NOP               Error return
NOP
```

Loader errors are now immediately available to the computer operator as the program is being run to assist in the correction and possible rerun if desired.

D. Linkage for Remote Display

The addition of a I/O subroutine for remote operation of a digital display unit (DD65) requires that proper entry linkages be entered into MCS. This will permit the driver to be called (if not within the MCS) and executed. These linkages are in the Available Equipment Table (AET) and the Available Equipment Driver Name Table (AEDNT). Both of these links

have been provided in the present (modified) version of the library tape. In addition, a reserve area of 1000 cells has been made available within MCS (cells 3060 to 4060) to contain the programming for the display driver.

The AET entries (in BOOT) are shown in figure 6 and the AEDNT entries are located at the end of the subroutine LOADER, at approximately cell 6460 in MCS. These entries are:

```
BCD     DISPLAY
SEV     3070
SLJ     HOWLER
```

where SEV indicates the driver is within MCS
and 3070 is the entry point address of the driver

This link therefore requires the display driver to have its entry point located at cell 3070 to be executed properly. If a new entry address is desired it must be entered into the AEDNT, the AET entry need not be changed.

The final modifications made to the COOP MONITOR system were the addition of two subroutines to the library tape. These subroutines were: 1. Grafplot, to allow a graph of any two designated arrays to be drawn on an incremental plotter, see section 4.1. 2. UCSD, a new Secondary Control System written by the University of California at San Diego, which allows the use of simple control cards for running "stacked job" input for compile and execution.

3.6 Specifications for Further Modifications

The modified version of COOP MONITOR is now ready for the next step in the development of an operating remote station capability. This is the programming and insertion into MCS of a remote station display (DD65) driver, the alteration of

the system interrupt handler (SELECT, REMOVE, DETECT) to allow remote station interrupts and a subroutine to display graphical information on the remote display. The specifications for these three modifications are discussed in the following sub-sections:

A. Remote Display Driver

This subroutine would allow the display to function as the comment-to-operator medium, normally the console typewriter. One of the subroutines for this driver has been completed and is fully described in section 4.2 "line printer simulator". This subroutine will produce a line by line display on the console (DD65), that will move upward with the arrival of each new line, like a typewriter. Only the upper half of the tube face should be used for the "line printer" to allow the lower half to be used as a "status board" for computer input or output assignments and any special information for the remote station operator.

To correctly process input/output requests from COOP MONITOR (from READ/WRITE) the remote display driver must accept the standard calling sequence and correctly respond to the parameters specified. The specifications for the standard calling sequence is now given in the COOP MONITOR format:

TITLE: DISPLAY

PURPOSE: To input from or output to the remote station display (DD65) in a format compatible with the COOP MONITOR.

USAGE: Calling Sequence;

```
            ENA         Location interrupt routine
L           RTJ         DISPLAY
            R.M.        E.C.
L+1         F.C.        A
            I           B
L+2                     Alternate Return ( error )
                        E
L+3                     Normal Return
```

R.M.= Recording Mode, 1 if format is BCD 8 char/word
2 if format is 8 typ. char/word
3 if format is 1 typ. char/word

BCD 8 char/word; implies the internal format is BCD with blanks replacing typewriter characters for which there is no BCD equivalent.

8 typ. char/word; implies the internal format is 8 typewriter characters per word.

1 typ. char/word; implies the internal format is 1 typewriter character per word.

E.C.= Equipment Code with format CRXX
C= Channel number (7 for display)
R= 1 for input, 0 for output
X= Not interpreted

I= Interrupt Selection
1= With interrupt, use the address in the A register
0= No interrupt, ignore the address in the A register

F.C.= Function Code explained below.
A, B, E = Detailed under each function code.

Functions: In all cases if a parameter error is encountered E is set equal to 0 and control is transferred to the alternate return. Parameter errors are such things as channel not equal to 7, R not equal to 0 or 1, incorrect function codes, etc. E is always set to reflect the case in which the typewriter (simulated by the display) is left. This has no meaning for the display but the correct response if necessary for COOP MONITOR. The 2^6 bit of E is set to 1 for upper case.

F.C.=1

Starts transmission of data to or from an area specified by the first address (A) and the last address plus 1 (B). In some cases a conversion and/or packing/unpacking operation is required by the recording mode.

For input, R.M.=1 (BCD 8/wd)

Keyboard input with no interrupt on carriage return is selected. Characters are packed eight per word until the area from A to B-1 is filled or carriage return is sensed. If carriage return is sensed fill out the last word with blanks if necessary. Interrupt on carriage return is selected and an exit made with the number of words read in the A register.

R.M =2 (Typ 8/wd)

This mode is the same as R.M.=1, except the BCD code from the display must be converted to typewriter characters.

R.M.=3 (Typ 1/wd)

The characters are read, converted to typewriter code, and stored one per word in area A to B-1. Interrupt on carriage return remains selected.

For output, R.M.=1 (BCD 8/wd)

The information from A to B-1 is packed into the proper format for the display and output. If the message overflows a three line buffer (186 characters) the overflow is lost.

R.M.=2 (Typ 8/wd)

The information from A to B-1 is converted to BCD and packed into the display format and output.

R.M.=3 (Typ 1/wd)

Same as R.M.=2 above.

F.C.=2 Check only

Control is returned with the number of words transmitted by the last read (R=1) or write (R=0) operation. This function if needed to satisfy MCS which assumes some of the transmissions are buffered and therefore it must return to the driver a second time to determine the number of words "buffered".

F.C.=3 Read/Write with checking

Same as the operation for F.C.=1, with the number of words processed entered in the A register before returning.

F.C.=4 Sense Equipment Ready

Determine if the display is in a "ready" status, if it is not ready set E to 1 and use the alternate return. If it is ready set E to indicate the current status for the "case" and use the normal return.

The present COOP MONITOR does not use any of the provisions incorporated for F.C. greater than 4. The driver routine need only check that the function code is within the range 1 to 4. If not set E=0 (parameter error) and use the alternate return.

B. Coop Monitor Interrupt Package

Before any great amount of programming can be done to adapt COOP MONITOR for remote operation, the present interrupt package (Select, Detect, and Remove) must be modified. This subroutine can presently select and sense interrupts on channels 1 thru 6 inactive and all internal interrupts. It cannot select or sense satellite generated interrupts on channels 3/4 or 5/6 via the CDC 1607 magnetic tape units, nor can it select or sense any channel 7 interrupt.

The calling sequence for the interrupt subroutine, see ref. 10, uses the integers 1 thru 12 to set the type of interrupt to be selected and sensed. The integers 1 thru 8 are used for those interrupts to be removed. These calling parameters need to be increased to provide the following additional options.

For Select:

Type Interrupt	Action
13	Select program control for channel 3/4
14	Select program control for channel 5/6
15	Interrupt on keyboard 1 hit (DD65)
16	Interrupt on keyboard 2 hit (DD65)
17	Interrupt on radar target hit (DD65)

For Remove:

To avoid confusion use the same parameters to remove the above selections

Before any of these changes are attempted the precautions necessary for programming an interrupt package, discussed in ref. 13, must be carefully considered.

Along with these modifications to the interrupt subroutine, the main frame of the computer should be modified to allow

computer selection of interrupt lockout. This would avoid the present procedure of selecting interrupt on arithmetic fault and then forcing a fault just to place the computer in lockout mode. There is just such a routine in MCS (RLOKOUT) to perform this function. This takes 20 instructions to accomplish what could be done with one. These modifications would be similar to those described in section 4.4 to provide computer selection of remote indicator lights.

C. Data Display Graph Plot Routine

The incremental graph plotting system has been operational for approximately six months, see section 4.1. A routine to provide this graph plot capability utilizing the Data Display unit (DD65), still needs to be completed. The following specifications and suggestions should provide an outline for incorporating this "Graph Display".

1. Use the author's scaling routine to scale to units per inch.
2. Use this scale and 16 increments per inch vice the 100 per inch used in the incremental system to build a scaled integer array of coordinate points.
3. Output the axis, titling, and labeling data using the standard "Grafplot" technique.
4. Curve drawing requires extensive use of the string capability of the display in order to provide for multiple curve graphs. This is due to the relatively small, 1024 cell 24 bit, memory of the data display.
5. Memory allocation in the DD65 Display should be about as follows:

    a. Vertical and Horizontal Axis, 8 inches long. Use vector size 64/line. Requires 40 cells.

b. Labels, 4 DD65 words/label. 18 labels requires 72 cells.

c. Titles, A minimum of about 200 cells.

d. Curves, Total memory usage of above is 312 cells. This leaves 714 cells for curves. Using the string vector mode this will provide storage for approximately 2500 vectors.

6. Using only the smallest vectors this will provide for approximately 20 traverses of the display tube. Assuming that an average curve traverses the tube 4 times, it appears that 5 curves may be displayed simultaneously.

Besides an on-line graph display routine, another routine which is very desirable would provide an off-line display of the graphs which were output by the CDC 1604 program for "hard copy" incremental plotting.

4.0 SYSTEM ROUTINES

Several system routines were written for both the Fortran 60 System and the Coop Monitor System. Four of these routines, which were developed, shall now be discussed.

4.1 Grafplot Routines

The 1604 Grafplot Routine, see enclouse 3, was written for the CDC 1604 computer. It generates a magnetic tape output in the format required to drive the 160A Grafplot Routine, see enclosure 2. A version of the 160A routine was written for the 160 computer, see enclosure 1. It bypasses the interpolation option in order to operate in the smaller memory of the 160 computer.

This system will produce multiple curved graphs, each curve is limited to a maximum of 900 points. Optional auto-scaling of data points, titling and curve labeling is available. Figure 7 is a general flow diagram of the 1604 Grafplot routine. Copies of the coding for these routines are available from the CDC DATA CENTER, PALO ALTO, CALIFORNIA or from the U.S.N. Postgraduate School Computing Center.

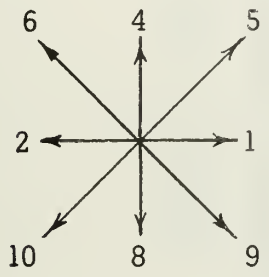
Two original techniques where developed which are of interest. They are the quantized scaling iterative procedure used in the 1604 program and the "decision rule" algorithm for providing "best fit" motion commands to an incremental plotter.

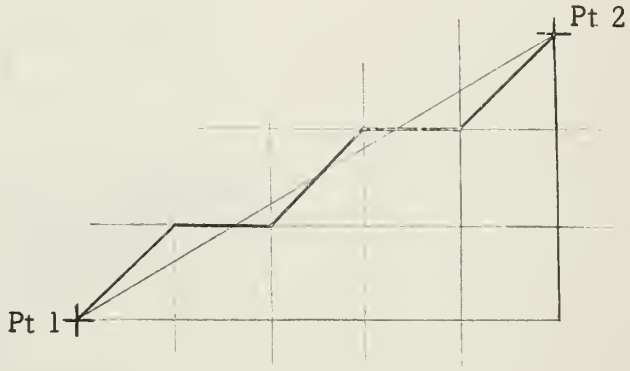
The scaling technique assumes a scale, tests for fit, then iterates to the next quantized scale and test again until the optimum available scale is found.

The two point connector algorithm, "LINE", computes the code string required to move the pen of a digital incremental X,Y plotter from an initial point (X1, Y1) to a terminal point (X2, Y2) by the "best fit" approximation to the straight line between the

points. The permitted elemental pen movement is to an adjacent point in a plane Cartesian point lattice, diagonal moves are permitted. The approximation is "best fit" in the sense that the deviation of the generated line from the true straight line never exceeds one half an increment. This is physically the best fit possible within the movement constraints imposed.

As an example of the LINE program, assume it is required to increment a pen's position with minimum deviation, along a straight line connecting point 1 and point 2. The movement codes for the CalComp Plotter are as follows;





Executing the LINE program, the following values are found;

A = 5
B = 3
NRLONG = 5
NRSHORT = 3
MOSTLY = 1
IBOTH = 5

and the iteration scheme produces the incremental path shown.

Note, in the following fortran program , LINE, a subroutine is required to output the movement codes from the user's particular computer to the incremental plotter. Thus subroutine "PLOT (MOVE)" would output the movement code "MOVE" to the plotter. For the CalComp Plotter the NCODE instructions may be deleted, since the routine is written to require no code conversion for this equipment.

```
      PROGRAM LINE ( X1, Y1, X2, Y2 )
      DIMENSION NCODE ( 10 )
      NCODE ( 1 ) = "user's +X  movement code"
      NCODE ( 2 ) =     "   -X      "       "
      NCODE ( 4 ) =     "   +Y      "       "
      NCODE ( 5 ) =     "   +X+Y    "       "
      NCODE ( 6 ) =     "   -X+Y    "       "
      NCODE ( 8 ) =     "   -Y      "       "
      NCODE ( 9 ) =     "   +X-Y    "       "
      NCODE (10) =      "   -X-Y    "       "
      A = X2 - X1
      B = Y2 - Y1
      IF ( A ) 200,100,100
  100 ICODE = 1
      Go To 300
  200 A = - A
      ICODE = 2
  300 IF ( B ) 500,400,400
  400 JCODE = 4
      Go To 600
  500 B = - B
      JCODE = 8
  600 IF ( A + B ) 1500,1500,700
  700 IF ( A - B )  800, 900,900
  800 MOSTLY = JCODE
      NRLONG = B
      NRSHORT = A
      Go To 1000
  900 MOSTLY = ICODE
      NRLONG = A
      NRSHORT = B
 1000 IBOTH = ICODE + JCODE
      LIMIT = NRLONG
      NSWITCH = NRSHORT
      MOSTLY = NCODE ( MOSTLY )
      IBOTH = NCODE ( IBOTH )
 1100 IF ( NRLONG - NSWITCH - NSWITCH ) 1200,1400,1400
 1200 MOVE = IBOTH
      NSWITCH = NSWITCH - NRLONG + NRSHORT
 1300 CALL PLOT ( MOVE )
      LIMIT = LIMIT - 1
      IF ( LIMIT ) 1500,1500,1100
 1400 MOVE = MOSTLY
      NSWITCH = NSWITCH + NRSHORT
      Go To 1300
 1500 END
      END
```

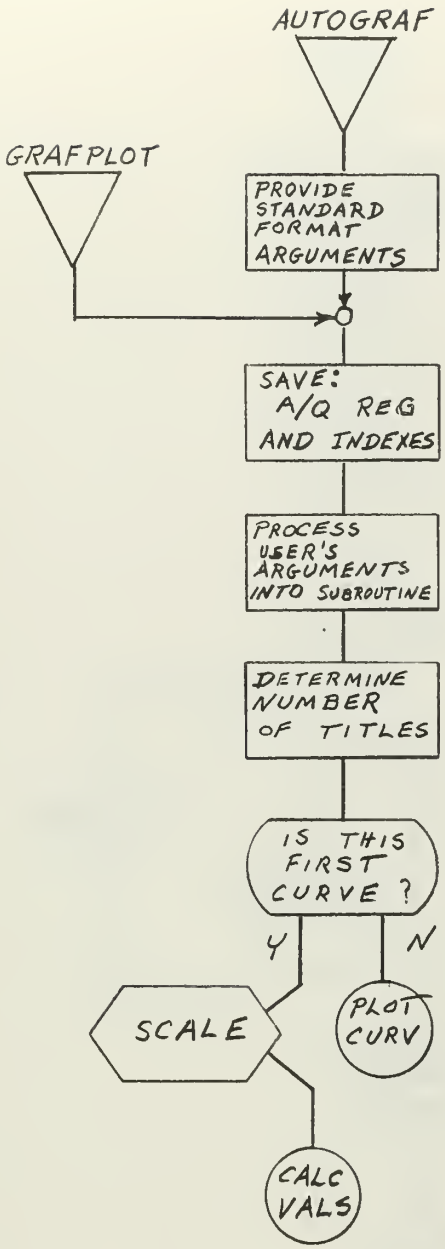

AUTOGRAF
GRAFPLOT
PROVIDE STANDARD FORMAT ARGUMENTS
SAVE: A/Q REG AND INDEXES
PROCESS USER'S ARGUMENTS INTO SUBROUTINE
DETERMINE NUMBER OF TITLES
IS THIS FIRST CURVE ?
Y
N
SCALE
PLOT CURV
CALC VALS

Figure 7a

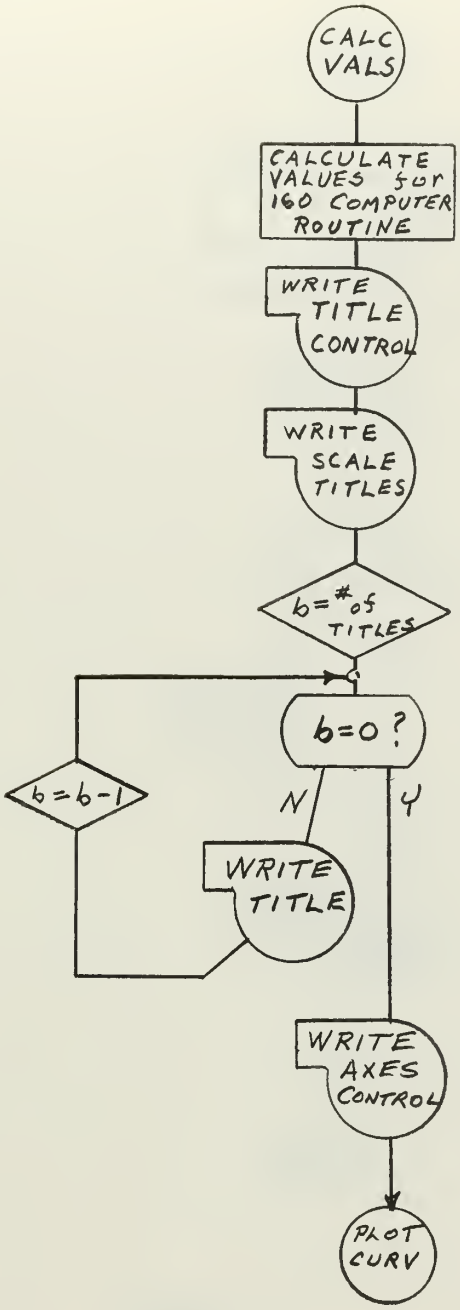

CALC VALS
CALCULATE VALUES for 160 COMPUTER ROUTINE
WRITE TITLE CONTROL
WRITE SCALE TITLES
b=# of TITLES
b=0 ?
b=b-1
N
Y
WRITE TITLE
WRITE AXES CONTROL
PLOT CURV

Figure 7b

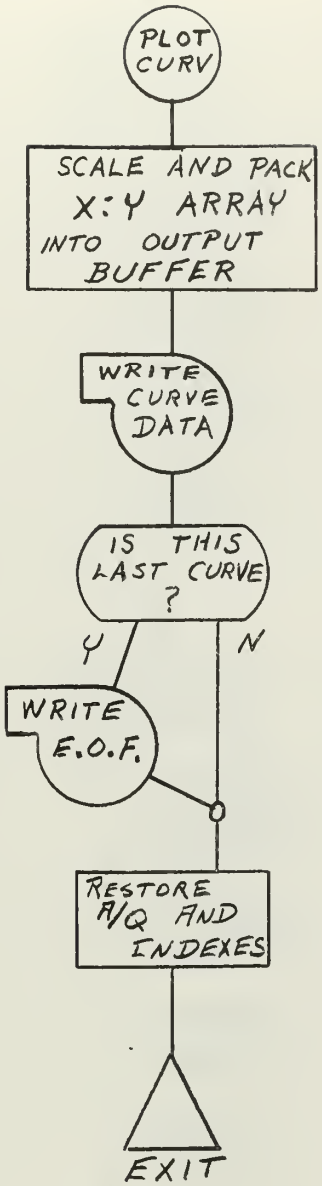

PLOT CURV
SCALE AND PACK X:Y ARRAY INTO OUTPUT BUFFER
WRITE CURVE DATA
IS THIS LAST CURVE ?
Y
N
WRITE E.O.F.
RESTORE R/Q AND INDEXES
EXIT

Figure 7c

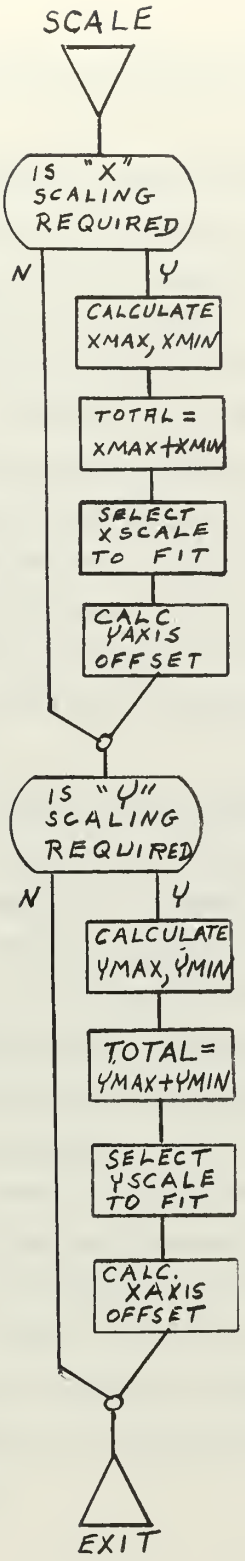

SCALE
IS "X" SCALING REQUIRED
N
Y
CALCULATE XMAX, XMIN
TOTAL = XMAX+XMIN
SELECT XSCALE TO FIT
CALC YAXIS OFFSET
IS "Y" SCALING REQUIRED
N
Y
CALCULATE YMAX, YMIN
TOTAL= YMAX+YMIN
SELECT YSCALE TO FIT
CALC. XAXIS OFFSET
EXIT

Figure 7d

4.2 Data Display "Line Printer Simulator"

The "line printer simulator" program demonstrates a very effective method of providing a continuous page listing ability. The problem was to display one full page of data on the Data Display Incorporated, Model 65, console. The data displayed was to be updated one line at a time by over-writing an old line of data in the DD65's memory. It was desired that this new line entry always occur at the bottom of the page and all other lines would precess up the page and finally disappear. An effective method to provide this listing ability was programmed and added to the Fortran 60 System. An understanding of this program requires a familiarity with the word format used to update the DD65. The DD65 has the following characteristics;

a. It has two keyboards which can be used only as output mediums to a computer. There is no connection or internal logic for providing tube display of a character when it's corresponding key is hit. This is different from a conventional typewriter, which prints it's character automatically when a key is hit. In order to display the character, it must be read by a computer, processed and the appropriate word format must be transmitted back to the display. The display logic then decodes the format of the words transmitted to its memory and displays the appropriate character/characters. There is no provision for read out from the DD65 memory by a computer. The memory serves only to store the data for display upon the cathode ray tubes. The logic cabinet of the display unit contains the circuitry required to scan its memory and display the contents onto the cathode ray tubes.

b. To read from a display keyboard, requires that a sequence of external function codes (see fig. # 9) must be executed to sense keyboard hit, select keyboard for input to the computer, then input one word via channel 7. The BCD code for the key read-in will be the lowest 6 bits for keyboard 1 characters and the lowest 9 bits if keyboard 2 was read. The process of reading the key into the computer releases the key at the DD65 console.

c. Output to the display maybe either characters or vectors. There are three sizes of characters, 32, 64, or 128 per line. There are two sizes of vectors, 64 or 128 per line. The line mentioned is the width of the tube display area, approximately 8.5 inches square.

d. A selection of either character or vector mode, a memory storage address, the X / Y display position, and a tube selection are all data which is conveyed to the display. The word which contains this information is called a designator word, see figure # 8. This designator word can also carry two characters or vectors, depending on the mode selected. Additional characters or vectors can be transmitted after the designator word, if packed in a format called "string word format", see figure # 8. These "string" word characters or vectors will be positioned relative to their preceding designator word. Thus it is the designator word that controls the tube location where the character/vector string will appear. For additional display information, see reference #19.

The line printer routine provides for displaying 32 lines of data. This requires that 32 designator words, each followed by a line of data, be stored in the DD65 memory. Once the data was in the DD65 memory, it was only necessary to selectively transmit new designator words with the appropriate new Y position indicators to move the whole line's position. Thus once the line data has been output, the 1604 is not required to retain an image of the page data in its memory, nor need any time be wasted in re-transmitting redundant information to the display.

The technique employed required that a 32 word designator table be constructed. This table provided memory addresses for data storage from cells 600 thru cell 1777 in the DD65 memory. Note, the DD65 memory cells are only 24 bits long, the disassembly from the 1604 computer's 48 bit word format is automatic in the DD65 logic. Note also that equipment limitations disallow addressing odd numbered memory locations in the DD65. The designator table also provides 32 discrete, equally spaced vertical "Y" axis locations to provide seperation between the lines to be displayed.

The method of up-dating the page displayed requires that all the "Y" locators in the designator words be incremented vertically one line space (20 units). A mask search is then executed to locate the designator word which contains the "Y" locator for the bottom line entry to the page. This "Y" locator is 400. Once found, the designator word is stored in the output buffer. Then the designator table is transmitted to the DD65, thus sweeping all lines vertically one space. The new line is packed into the output buffer. This buffer now contains a

designator word followed by a line of string words. These are now output via channel 7 to the display's memory. Note the channel 7 input / output instructions transmit words, from the 1604 memory in reverse sequence compared to the normal buffer channels. This requires an inverse packing order when packing the buffer for output on channel 7.

This line printer technique has also been incorporated into a typewriter page simulator for the right tube of the display. Both systems are working well.

See appendix IV for the resident coding required for the line printer package. It required only 77 cells including the designator table and a 10 cell output buffer. The designator table is listed in the appendix IV, page # 6.

CHANNEL 7 OUTPUT FORMAT for DD65 DATA DISPLAY

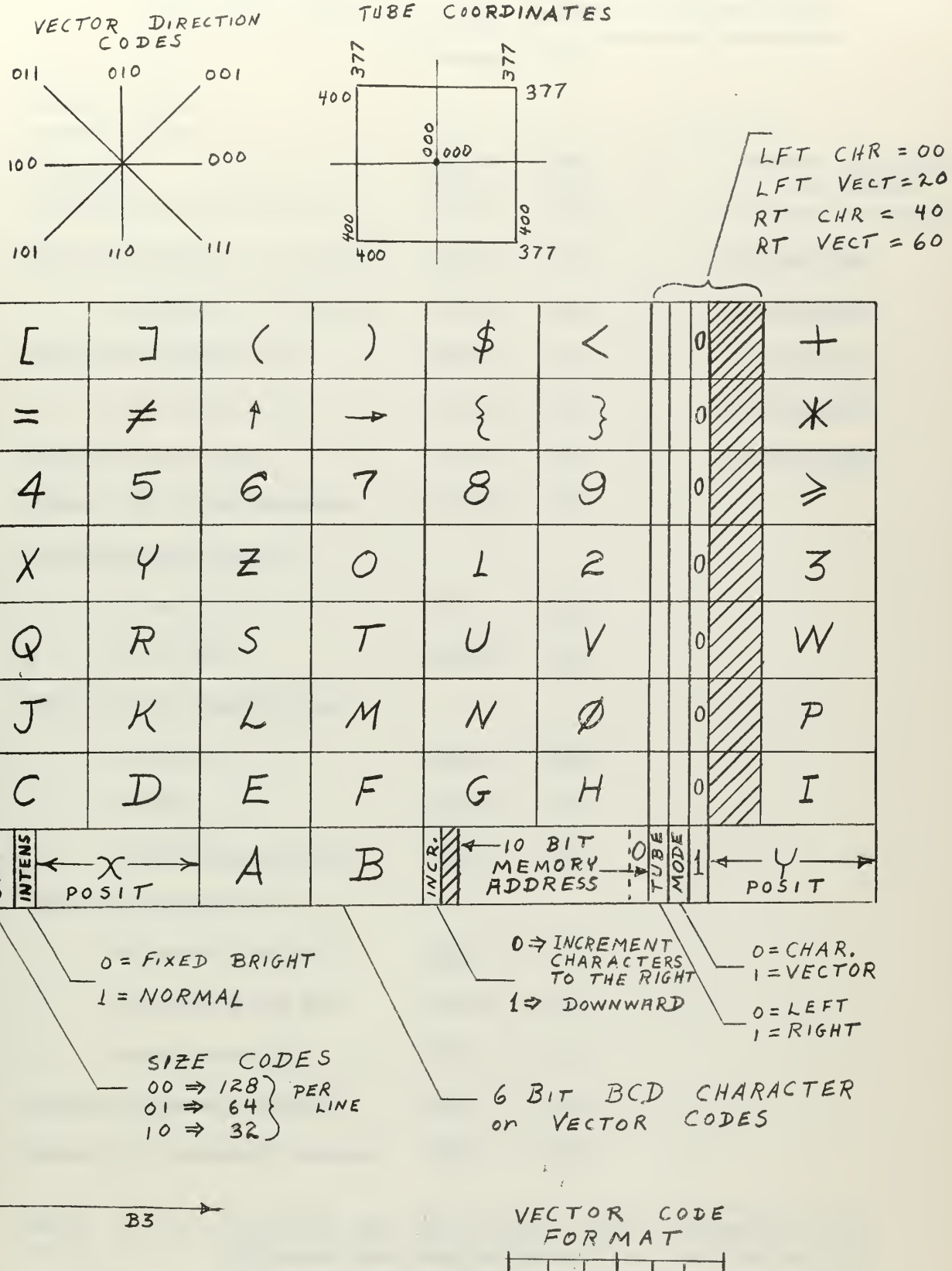


Figure 8

DATA DISPLAY CONSOLE MODEL 65 ''EXTERNAL FUNCTION CODES''

10 December 1962

SELECT CODES

FUNCTION	Select Code 1604	Code 160	Computer Input Word Bit Format
Select Keyboard 1 for Input	[1]C7140	7140	000000FFFFFF
Keyboard 2 for Input	C7120	7120	000VVVFFFFFF
Select Track Ball ''X''	C7102	7102	000XXXXXXXXX
Track Ball ''Y''	C7104	7104	000YYYYYYYYY
Select Range Switch	C7110	7110	000000000RRR
Select 160 Status Response	[2](C7040)	7040	
Select Memory Update;			
from 1604	C7010	(7010)	
from 160	(C7020)	7020	
Select Radar Target Data;			
to 1604	C7002	(7002)	
to 160	(C7004)	7004	
to Auxiliary Equip.	C7001	7001	
Select Interrupt on;			
Keyboard 1 Key Hit	C7105	(7105)	
Keyboard 2 Key Hit	C7103	(7103)	
Radar Pulse Hit	C7141	(7141)	
Release Interrupt Request	C7111	(7111)	
Remove All Interrupt Selects	C7121	(7121)	

Notes
1. ''C'' Indicates 1604 data channel number. (Normally, C=7)
2. ( ) Indicates code disabled selectively on ''160 or 1604 ONLY'' mode switch position.

Figure 9a

SENSE CODES

CONDITION	1604 Full Exit Sense Code	160 Status Response
Keyboard 1 Hit	C7172	xxx2
Keyboard 2 Hit	C7175	xxx1
Keyboard 1 NOT Hit	C7173	
Keyboard 2 NOT Hit	C7174	
Tab Hit		xx1x
Tab NOT Hit	C7157	
Carriage Return Hit		xxx4
Carriage Return NOT Hit	C7166	
Keyboard 1 Selected		xx4x
Keyboard 2 Selected		4xxx
Keyboard 1 NOT Selected	C7167	
Keyboard 2 NOT Selected	C7037	
NOT DD-65 Interrupt	C7156	
DD-65 from 160 Selected	C7020	2xxx
DD-65 from 1604 Selected	C7010	1xxx
Radar to 160 Selected	C7004	x4xx
Radar to 1604 Selected	C7002	x2xx
Radar to Auxiliary Equipment NOT Selected	C7001	x1xx
Radar Target Present	C7136	xx2x
Radar Target NOT Present	C7137	

Figure 9b

4.3 Merg Sort

The SORT Routine, see appendix III, was written to improve the effeciency and reliability of sorting using the 1604 computer vice the IBM mechanical card sorter. Another motivation was to provide the capability of sorting 120 character records which could not be easily sorted using 80 character cards.

The SORT Routine can sort on variable length column fields from 1 to 120 columns. The present routine divides the field to be sorted into 7 character fields, then multiple entries are made to a subroutine, SORT7, which actually performs the sorting operation. SORT7 divides the computer's memory into three equal bins, ABIN, BBIN, and CBIN. These are each 20,000 octal cells long. SORT7 requires an input tape written in 120 character records with an "End of File Mark" terminating the file to be sorted. Upon entry to SORT7, one of its subroutines, Alsinp, reads 1000 octal cards into ABIN. As each card is read, a sorting key is formed from the column field designated and this key is stored in the cell preceeding the input record image from which it was generated. The key occupies one cell, the record image occupies 17 octal cells, thus each input record requires 20 octal memory cells. Multiply 20 octal by 1000 octal to obtain the arbitrary 20,000 octal bin size which was choosen as the basic block size. Alsinp always checks for an E.O.F. mark and sets a flag to signal the rest of the routine that the last block has been read. After Alsinp has read 1000 octal records or encountered an E.O.F. mark, then an entry is made to a subroutine, Rlhmerg, which performs an internal merg sort by oscillating between ABIN and BBIN, sorting on the keys. The internal merg sort terminates with the sorted block in CBIN. If this was the first block, it is immediately written onto one of the two scratch tapes and Alsinp inputs another block. If this was not the first block, then a subroutine, Hexsort, is entered to perform a merg of the new block

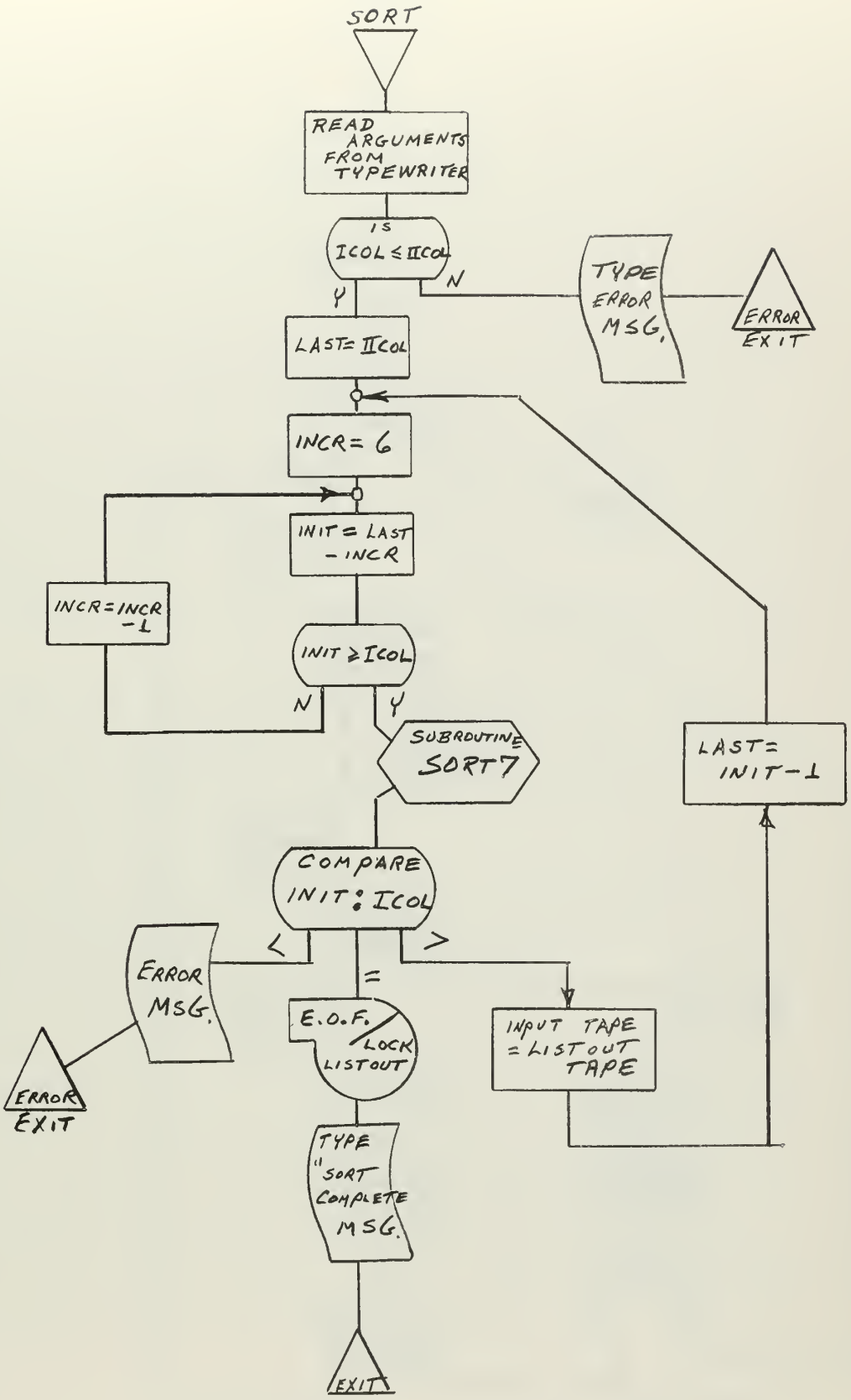

SORT
READ ARGUMENTS FROM TYPEWRITER
IS ICOL ≤ IICOL
N
TYPE ERROR MSG.
ERROR EXIT
Y
LAST = IICOL
INCR = 6
INIT = LAST - INCR
INCR = INCR - 1
INIT ≥ ICOL
N
Y
SUBROUTINE SORT7
LAST = INIT - 1
COMPARE INIT : ICOL
<
=
>
ERROR MSG.
ERROR EXIT
E.O.F. / LOCK LISTOUT
INPUT TAPE = LISTOUT TAPE
TYPE "SORT COMPLETE MSG.
EXIT

Figure 10a

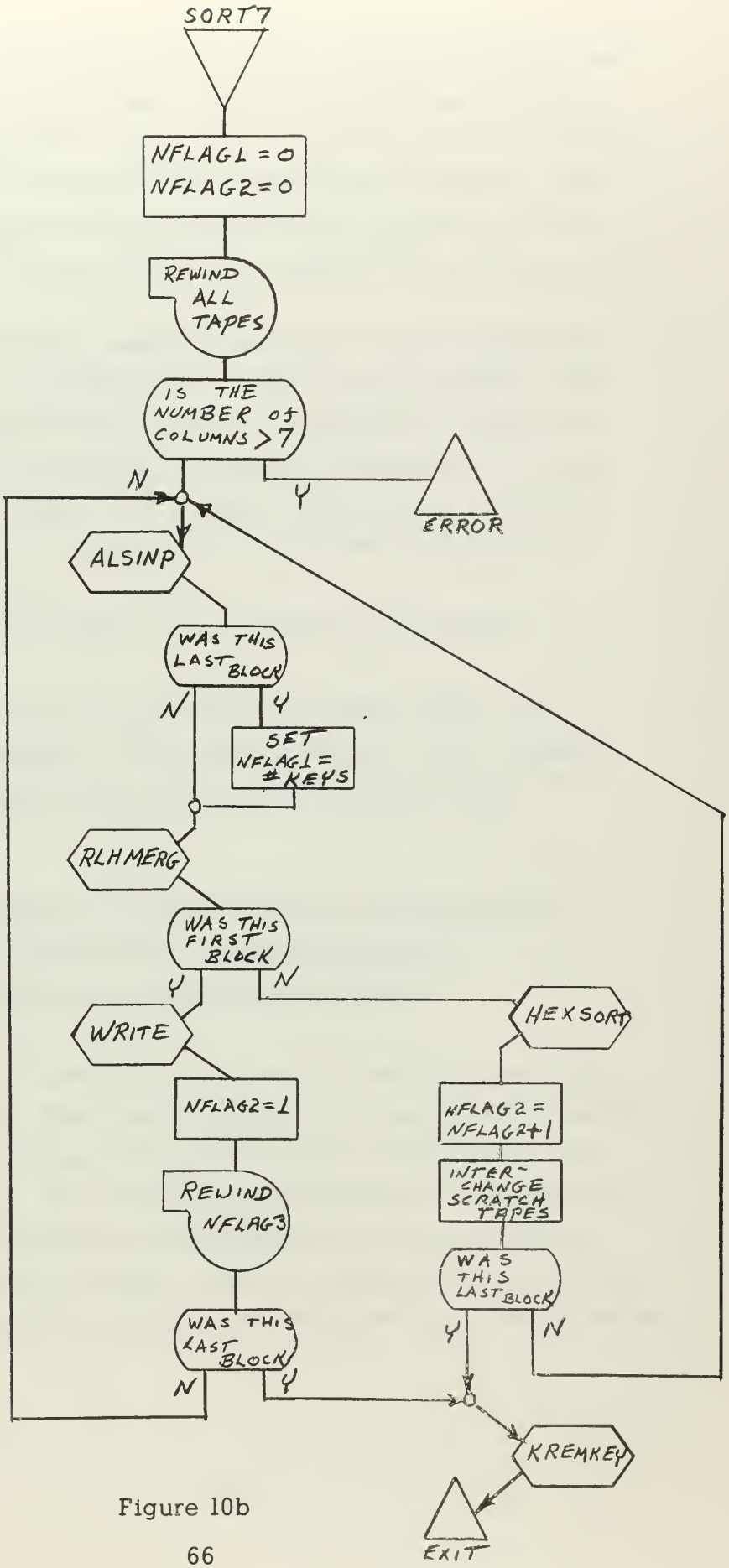

SORT7
NFLAG1=0
NFLAG2=0
REWIND ALL TAPES
IS THE NUMBER OF COLUMNS >7
N
Y
ERROR
ALSINP
WAS THIS LAST BLOCK
N
Y
SET NFLAG1= #KEYS
RLHMERG
WAS THIS FIRST BLOCK
Y
N
WRITE
HEXSORT
NFLAG2=1
NFLAG2= NFLAG2+1
REWIND NFLAG3
INTER-CHANGE SCRATCH TAPES
WAS THIS LAST BLOCK
WAS THIS LAST BLOCK
Y
N
N
Y
KREMKEY
EXIT

Figure 10b

into the file of previously sorted blocks which are accumulated externally on a scratch tape. This cycle continues until an E.O.F. is encountered, then after the external merg is complete, the scratch tape is rewound and a subroutine, Kremkey, reads in the ordered blocks from the scratch tape, removes the keys and writes out the output file in an ordered, 120 character record length file.

If additional fields remain to be sorted, this output file is designated the input and another entry is made to SORT7 with the new character field's initial and last column numbers as arguments. This procedure continues until the total sort field is in order and listed on the output. The output is then rewound to lockout and the operator is informed that the job is finished.

Several improvements to this routine are suggested.

a. Build and sort on a two cell key, if large fields are to be sorted frequently. This would provide a better balance between the internal sorting time and the external tape handling time.

b. Provide for sorting on split fields, ie. non-consecutive column fields. This could be provided by forming the sort key piece-wise from specific columns.

c. Use a double entry key / key location table for performing the internal sort. This would eliminate the necessity of carrying the input record image around with the key each time the key order is changed. The sort could be performed by ordering only the keys and maintaining correspondence in the key location table as the keys are sorted. This key location table could then be used to perform the internal ordering of the input record images.

This scheme would improve the internal sorting speed and reliability since it would require fewer internal data transfers.

d. The last and probably the most significant improvement which is indicated, would be to utilize two more scratch tapes. This would improve the sort routine speed by nearly 50%. This is due to the rewind time required by the tape drives and their inability to read in the reverse direction. At present, much time is wasted waiting on the two scratch tapes to rewind in order that the next internal block can be merged with the external ordered file. A suggested method to improve the external sort would be to alternate the merging of the internal block between to sets of scratch tapes similar to the set presently employed. This would increase the rewind time available for each set and thus decrease the idle time mentioned previously. After Hexsort has finished merging the last block, then Kremkey would have to operate differently than at present. Kremkey would have to read one block from the ordered tape of each scratch tape set, then perform a final merg comparison similar to the Hexsort merg routine. Instead of forming an ordered list in memory, this final merg could be written out as the final ordered, 120 character listable output. Before any other modification is considered, this last suggestion should be given priorty "one". At present it remains a gross deficiency against high speed sorting using this routine.

4.4 Satellite System Programming

There are two basic operations necessary to establish an adequate satellite system. These are the ability to exchange data and to selectively execute a pre-stored program when directed by another member of the satellite system. The present installation proposed to couple the CDC 160 to the CDC 1604 Computer. The CDC 160 Computer does not provide an interrupt mode, therefore only the 1604 interrupt mode was available to be employed.

The desirable characteristics of the basic 1604 satellite interrupt program are:

a. Reliability
b. Flexibility
c. Speed
d. Effective Employment of the Communication Flags
e. Minimize Memory Requirements

The problems encountered in designing the basic 1604 interrupt routine shall now be discussed. The 1604 Computer is unable to prevent the 160 Computer from entering the interrupt auxiliary routine which is wired into the 1604. For this reason, a protective "lock-out" scheme was required to prevent interference with the 1604 equipment handling routines. Specifically, there exists a critical time interval between equipment selection and activation during which an interrupt from a satellite station would be disastrous to the main program. The technique develope to prevent this, is as follows: All 1604 equipment handling routines were modified to turn on communication flag I (see ref. # 2) in advance of any equipment selection. Communication flag I is then cleared after the selected equipment is activated. Using this scheme it is

then required that the 160 satellite computer inspect communication flag I prior to executing an interrupt. If flag I is set, then the 160 computer enters a "wait loop" until flag I is cleared by the 1604 Computer. Thus through the use of communication flag I, the 1604 computer is, in effect, capable of "locking out" the 160 satellite computer from an interrupt request. This was a desirable feature not presently provided in the hardware design.

The next problem in the design of the interrupt routine was how to indicate to the 1604 computer, the nature of the task requested by the 160 computer. The method used requires that each interrupt shall immediaty trigger a , 48 bit word , data transfer to the 1604. The format of this word, designated as the "control word", is outlined in detail in figure # 12. After receiving this "control word" the 1604 interprets the operation to be performed. There are four tasks which can be directed by the control word, they are:

1. 1604 INPUT DATA BLOCK
2. 1604 OUTPUT DATA BLOCK
3. PACK ARGUMENT ADDRESSES INTO A SUBROUTINE AND THEN EXECUTE THAT SUBROUTINE.
4. LOAD "A" REGISTER WITH A 21 BIT ARGUMENT FROM THE CONTROL WORD AND THEN RETURN JUMP TO EXECUTE THE SUBROUTINE.

A flow diagram of this 1604 satellite interrupt routine is provided in figure # 11.

One additional problem remains to be discussed, that is storage allocation for the programming required by the satellite system. This was acquired by setting the resident bias level on the Fortran 60 System to 5000 octal, vice the normal 4000.

1604 INTERRUPT ROUTINE
FLOW DIAGRAM

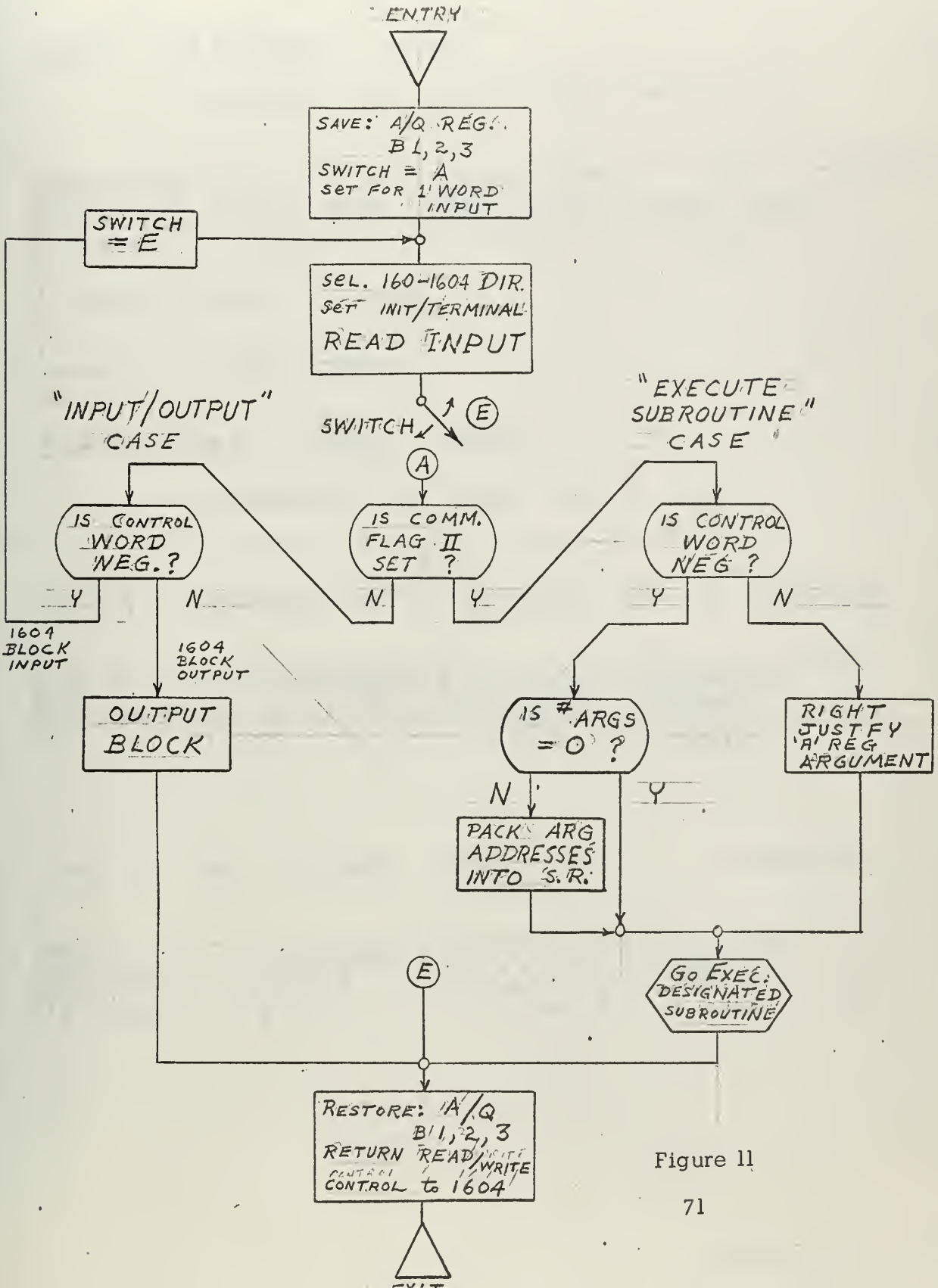
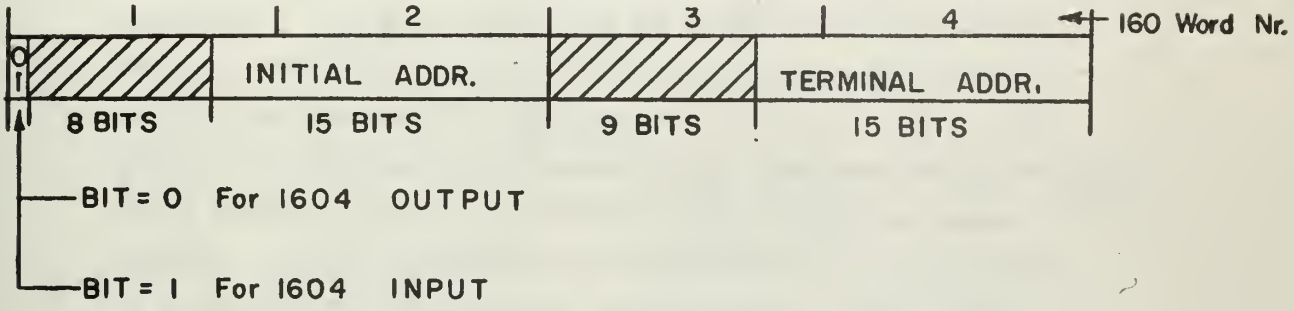


Figure 11

160-1604 SATELLITE CONTROL WORD FORMAT

DATA TRANSFER MODE

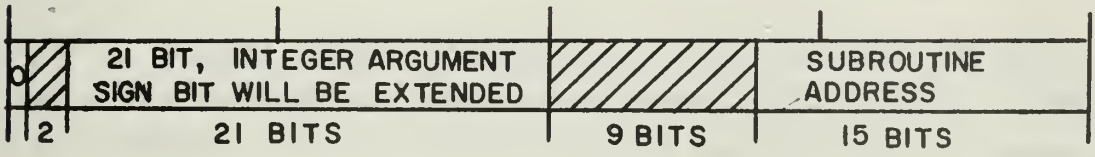
INTERRUPT WITH COMM FLAG II "NOT" SET



SUBROUTINE JUMP MODE

INTERRUPT WITH COMM FLAG II "SET"

TYPE I, ARGUMENT IN "A" REGISTER, RTJ To SUBROUTINE



TYPE II, PACK ARGUMENT ADDRESS, RTJ To SUBROUTINE

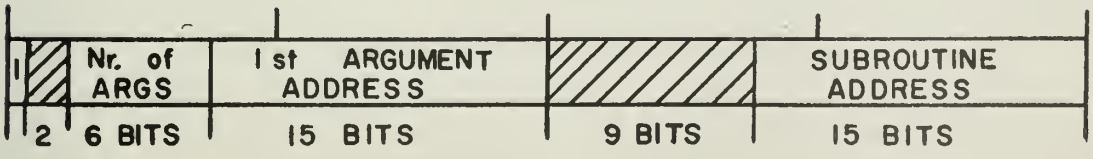


FIGURE 12

This provided the memory space needed to store the satellite programming plus an additional "buffer" area for receiving transient subroutines from the 160 computer. These subroutines may the be executed by an interrupt entry to the 1604 computer via the satellite interrupt system.

The use of the 160 as a satellite station requires that the modified library tape, FORTSAT, be in use by the 1604 computer. Also, the 1604 must be operating under monitor control in order to utilize a remote monitor process routine which was developed by the authors' (see ref. # 22). The physical seperation of the satellite 160 computer station required some means of sensing the above conditions at the remote station. Visual indicators were selected as the best method of implementing the required status sensors. The console of the 160 Computer contains many unused indicator lights, two ot these were selected to be used as status indicators. The first light, designated # 1 or FORTSAT, is a blue-green background field light at the left end of the "P" register. The second light, designated # 2 or MONITOR, is the next blue-green background field light to the right of the # 1 light.

Manual control of these lights at the console of the 1604, was considered an unnecessary burden to the operator. Computer selection would be more convenient and the status indications more current in the reflection of the actual status of the 1604 Computer operations.

To enable computer control of the indicator lights, two new select codes (on - off pairs) for the 1604 were required. As all satellite selects are made in the 1607 Magentic Tape Logic Cabinets, modification of these units was necessary. This modification would allow the 1607 to accept and decode the new indicator light codes. The two codes selected were;

LIGHT # 1	(FORTSAT)	"ON"	53511
		"OFF"	63511
LIGHT # 2	(MONITOR)	"ON"	53544
		"OFF"	63544

The select code format is as follows:

First digit;	Channel number
Second digit;	Cabinet number
Third digit;	Group code, 5 for satellite
Fourth & Fifth;	Condition code

The two codes selected for the indicator lights contain the false cabinet number 3. This was necessary because the 1607 (cabinet # 2) does not uniquely decode the last two digits of any select code. Therefore, to prevent interference with any of the existing codes, a false 3 was used.

To avoid the need for manual selection of "program control" mode of operation for the 1607 Tape Cabinets, when enabling the satellite station, this selection was wired to be controlled by light # 1 and the master clear switch. Initial turn on of light # 1 selects "program control" and the Master Clear disables both lights and removes the program control selection. The logic diagrams for the modificiations made to the 1607 Tape Cabinets are shown by figure # 13. Two relays were added to the 160 to provide switching of the two lights provided.

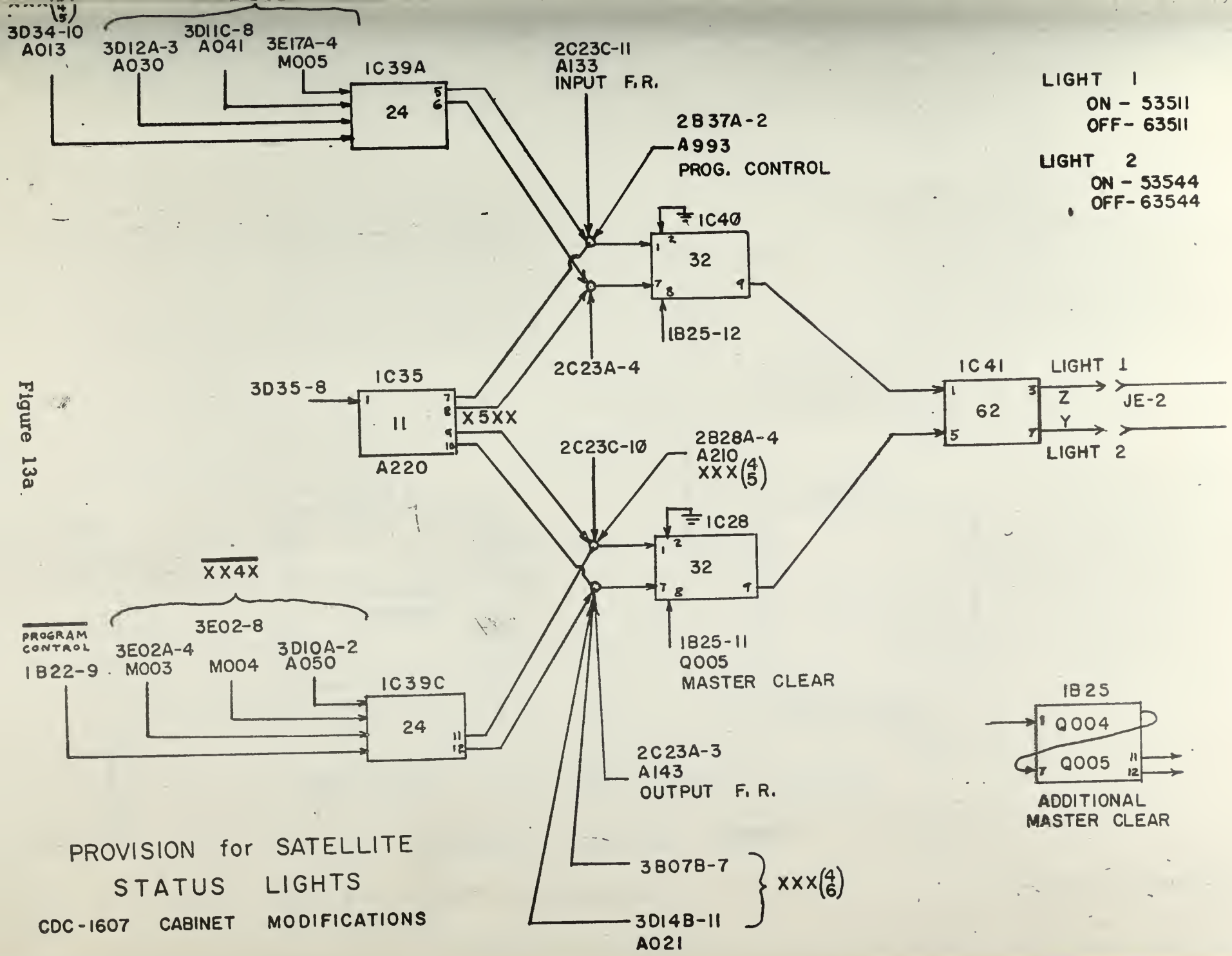


Figure 13a

PROVISION for SATELLITE
STATUS LIGHTS
CDC-1607 CABINET MODIFICATIONS

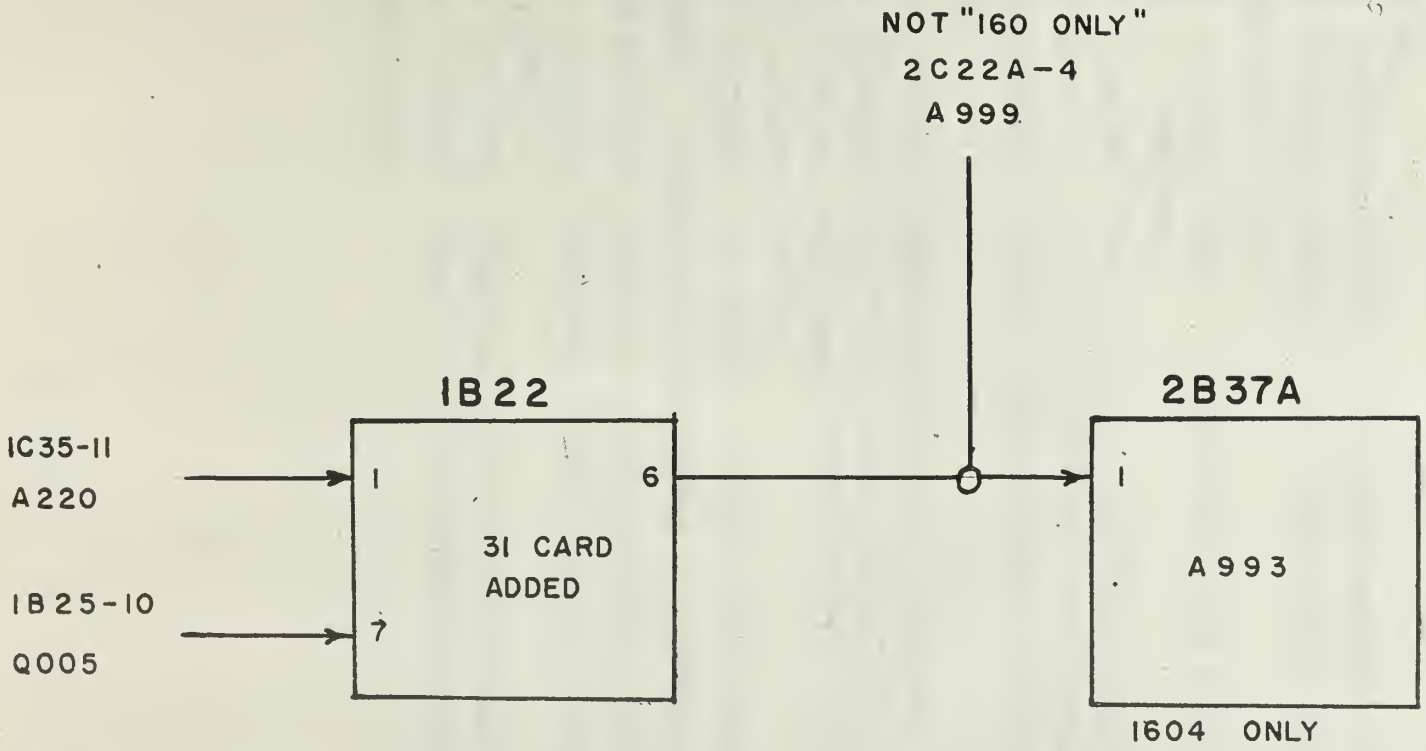


Figure 13b

PROVISION for EXF CODE CONTROL

of "PROGRAM CONTROL MODE"

CDC 1604 CABINET MODIFICATION

BIBLIOGRAPHY

1. CDC MANUAL, Pub# 025, INPUT-OUTPUT SPECIFICATIONS
2. CDC MANUAL, Pub# 187, PROGRAMMING MANUAL FOR CONTROL DATA SATELLITE COMPUTER SYSTEM
3. CDC MANUAL, Pub# 023, 160 COMPUTER PROGRAMMING MANUAL
4. CDC MANUAL, Pub# 167, 1604 COMPUTER PROGRAMMING MANUAL
5. CDC MANUAL, Pub# 032, 1604 MAINTENANCE MANUAL , Vol 2
6. CDC MANUAL, Pub# 037, 1607 DESCRIPTION AND OPERATION, Vol 1
7. CDC MANUAL, None , MAGNETIC TAPE UNIT (1607) CIRCUIT DIAGRAMS
8. CDC MANUAL, Pub# 069, 160 MAINTENANCE MANUAL, Vol 2
9. CDC MANUAL, Pub# 509, COOP MONITOR/OPERATOR'S GUIDE
10. CDC MANUAL, Pub# 508, COOP MONITOR/PROGRAMMERS GUIDE
11. CDC MANUAL, Pub# 506, FORTRAN 62/REFERENCE MANUAL
12. CDC MANUAL, Pub# None, CODAP1/REFERENCE MANUAL
13. CDC MANUAL, Pub# None, SPECIFICATIONS FOR THE COOP MONITOR SYSTEM
14. CDC BULLETIN, PSB-AS06621, GUIDE FOR CONVERTING FORTRAN 60 PROGRAMS TO FORTRAN 62 PROGRAMS
15. CDC BULLETIN, PSB-AE01, COOP MONITOR USAGE
16. CDC BULLETIN, PSB-AE02, COOP MONITOR MANUALS SUPPLEMENT
17. CDC BULLETIN, PSB-AE04, COOP MONITOR LIBEDIT ROUTINE
18. CDC BULLETIN, Pub# 516, COOP MONITOR/LIBRARY SUBROUTINES

19. C.G. LAWSON, AN INTEGRATED DISPLAY AND CONTROL SYSTEM FOR MAN-MACHINE COMMUNICATION, U.S. NAVAL POSTGRADUATE SCHOOL M.S. THESIS, 1962
20. R.P. WARRICK, DESIGN OF AN AUTOMATIC MULTI-PROCESSING SYSTEM, U.S. NAVAL POSTGRADUATE SCHOOL M.S. THESIS, 1962
21. CDC MANUAL, Pub# 087A, FORTRAN 60 SYSTEM MANUAL
22. R.L. HOGG/D.C. GLOVER, PROGRAM CONTROL SYSTEM FOR A SATELLITE MODE COMPUTER COMPLEX, U.S. NAVAL POSTGRADUATE SCHOOL REPORT # TR-DCL-62-11/13, 21 DECEMBER 1962

APPENDICES

APPENDIX I

LIBEDIT PROCEDURE

There are two publications available to assist a user of COOP MONITOR in performing a successful library preparation or modification. These are ref.17 and 10. The following was obtained from these two sources together with additional remarks by the authors to aid in the understanding of the procedure.

A. Library Tape Format

The library tape is composed of two files. The first file contains three records in absolute binary format. These are:

a. Bootstrap routine (BOOT)
b. Recovery routine (EXTREC)
c. Master Control routine (MCS)

The second file contains the subroutine library and starts with the "directory". This is followed by the library subroutines each consisting of one record (the system can accept subroutines with more than one record but the present library has none) composed of contiguous relocatable binary card images. Each subroutine is identified and defined in the directory by the following.

A subroutine name
Number of records in the subroutine (normally 1)
Number of common blocks assigned
Names of those common blocks
Length of each common block
Range of the subroutine (FWA and LWA+1)
Number of entry points to the subroutine
Entry point names
Number of external symbols referenced
External symbol names

The order in which the subroutines are identified in the directory must be the same as the order of appearance of the subroutines in the file. The identifying name, each entry point name, each external symbol and each common block name must have the following properties:

a. Eight BCD characters in length, no imbedded blanks
b. Leading character not digital or blank, except common block

The first library subroutine defined in the directory is the Master Control System called "Resident". The length of Resident and all named entry points are defined. External references are not made within the resident program. There are no entries in the directory for the subroutine EXTREC.

The recommended layout of the library subroutine file is:

1. Directory (this must be first)
2. I/O drivers for standard medium
3. All other I/O drivers
4. Secondary Control Systems (SCS)
5. Any other subroutines
6. SNAP (selective dump routine)
7. MAP (produces a memory map of loaded programs)
8. Systems such as FORTRAN, CODAP1, ALGOL, etc.

This layout facilitates one pass loading of library subroutines. The sequence for loading library subroutines will be:

1. I/O drivers for the standard mediums.
2. I/O drivers for the current job.
3. The desired secondary control system.

B. Directory Format

The directory is a set of 8 different types of cards for each subroutine in file 2 of the library tape. These must be ordered in the directory deck in the same sequence as the routines appear

in file 2. Care must be exercised to assure the directory reflects the exact composition of the subroutine file, otherwise errors will result. Each set of directory cards must contain:

1. A card with the subroutine name, left-justitied, in columns 1-8.

2. A card with a 16 digit octal integer, left-justified, in columns 1-16. This has the format:

000KKKKK000RRRRR

where 000 is zero

KKKKK is the number of common blocks required by the subroutine

RRRRR is the number of records the subroutine occupies on the library tape (usually 1)

3. If KKKKK is not zero, there must be a number of cards equal to KKKKK containing the name of each common block. The name is punched, left-justified, in columns 1-8. The card may be blank if no name for the common block is given.

4. If KKKKK is not zero, there must be a number of cards required to define the lengths of the common blocks. These cards contain octal integers separated by commas. The integers specify the length of the common blocks named by the cards in #3 and the lengths must be in the same sequence as the names. Any number of integers, more than zero may appear on a single card. The last integer on the card must be followed by a blank. The number of cards required depends on the number of common blocks and the packing of lengths.

5. A card with a 16 digit octal integer, left-justified, in columns 1-16. This has the format:

PPPFFFFFXXXLLLLL

where PPP is the number of entry points to the subroutine (must not be zero)

FFFFF is the relative first word address of the subroutine (FWA)

XXX is the number of external symbols referenced by the subroutine

LLLLL is the relative last word address plus 1 of the subroutine (LWA+1)

6. The number of cards equal to PPP, each containing an entry point name, left-justified, in columns 1-8. There must be at least one such name.

7. The number of cards required to define the entry point addresses. These cards contain octal integers of 1 to 5 digits separated by commas. The integers specify the relative address of the entry points named on the cards in 6. The entry point addresses must be in the same sequence as their corresponding entry point names.

8. The number of cards equal to XXX in 5. If this is greater than zero, each card just contains the external symbol name, left-justified, in columns 1-8. These external symbol names must be in the same order as they were defined during the assembly or compilation of the subroutine.

The sets of directory cards are preceded by a card with "DIRECTRY" punched in columns 1-8 and followed by a card with "7777777777777777" punched in columns 1-16.

C. Edit Input

The input data for editing a library tape is composed of control cards, directory cards, and program information. The program information is in the form of relocatable binary cards and this binary format must be thoroughly understood by the user to perform a successful LIBEDIT. Appendix III of ref.10 gives a complete description of this binary card format.

The input data must be on the standard input unit along with the required MCS control information ("begin job" and "mcs" cards). The next card of the edit input deck must

LIBEDIT INPUT DECK

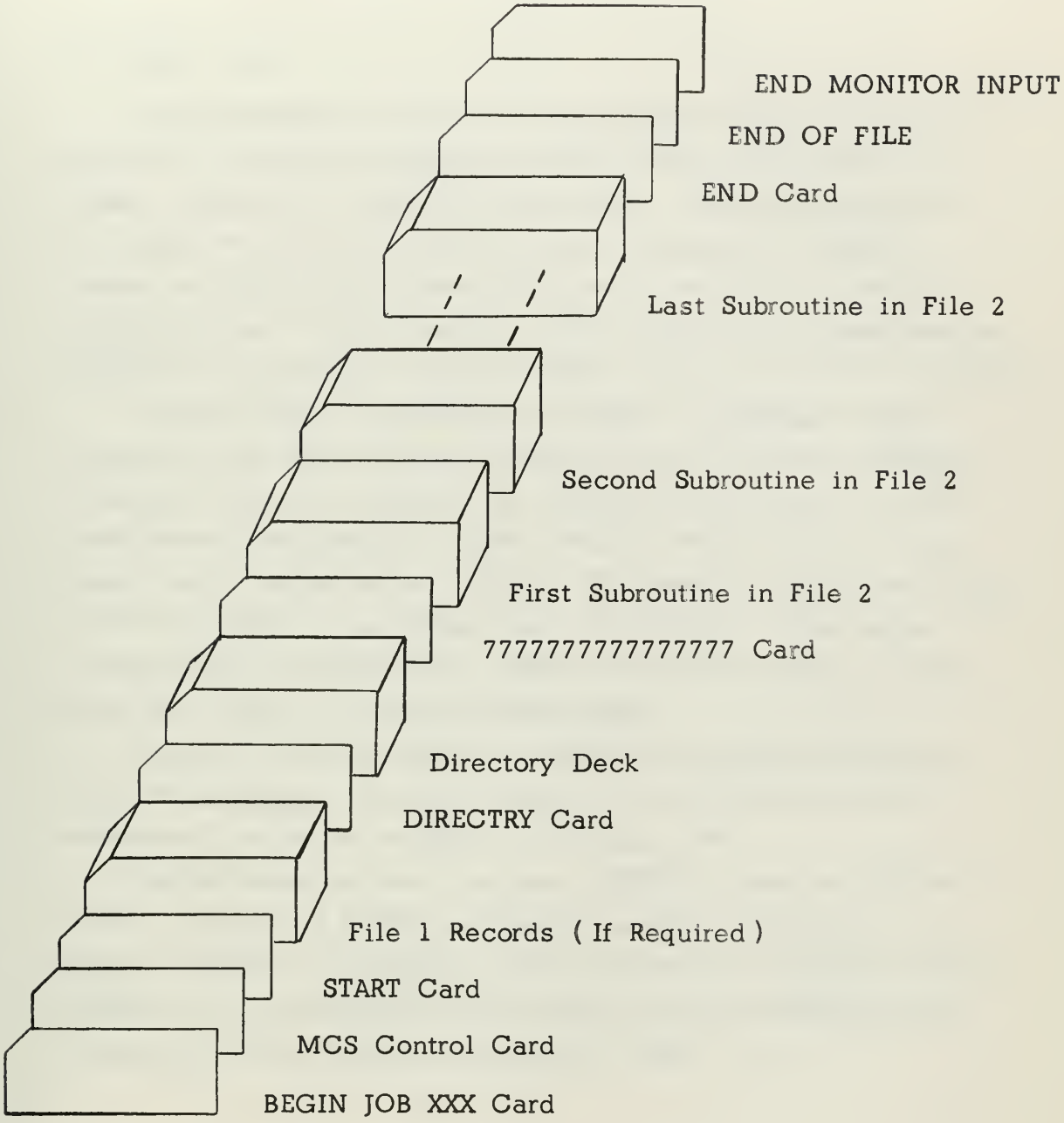


Figure A.2 Edit Input Deck

be a START card. This card contains the word "START" in columns 10-14 to produce a library tape of the same format as the old, see ref.17 for other formats possible. The succeeding cards in the input deck are determined by the type of edit and the file of the library affected.

D. File 1 Edit

The information in file 1 is in absolute binary format, and the file is divided into three records. Each record must be added, replaced or deleted as an entity. The information to be added or replaced must have been prepared by CODAP1 using the ORG pseudo instruction and it must be a complete, absolute binary format program. The complete deck produced by CODAP1 is included (only the "begin job" card need be removed).

The edit input is composed of decks of absolute binary records, each preceded by a control card. This card contains a decimal integer in column 10, followed by a blank, R, or D. The integer specifies the record position within file 1 of the binary card images which follow. The blank specifies the record is to be added, R to replace, and D to delete. The decks composing the records must appear in the same order in which they will appear on the new library tape.

If any of the entry points in MCS are altered by this edit then the complete directory (with the new MCS entry point addresses) must be included in the edit input.

The Extended Recovery routine (EXTREC) uses the AET and flags contained in BOOT as absolute address locations. If these locations are changed, then they must be changed within EXTREC (by recompiling) and including the new EXTREC deck, between BOOT and MCS, in the file 1 edit.

E. File 2 Edit

The information in file 2 is in several records in relocatable binary format. The first record is the directory. A new version of the directory need not be included in the edit input deck when the changes to file 2 are replacements which do not affect record lengths or entry points. In all other cases the complete directory must be included.

The first record of each subroutine (normally there is only one record) must be composed of relocatable binary card images produced by CODAP1. For edits involving subroutines of more than one record see ref.17. Each record must be preceded by a control card containing the subroutine name, left-justified, in columns 1-8, a blank in column 9, and a decimal integer starting in column 10 followed by a blank, R, or D. This integer specifies the position of this record within the records of this subroutine (usually 1). The blank specifies that the record is to be added; R specifies it is to replace an existing one, D specifies it is to be deleted. If a D appears, a binary deck for that record must not appear in the edit deck.

The edit deck, see figure A.2, must be terminated by a card with the word END in columns 10-12. To finish the input job an "END OF FILE" and a "END MONITOR INPUT" are necessary.

F. Operation

The library edit program is called from the library as the Secondary Control System by a MCS control card. The edit input is read from the standard input unit; a new library tape is written on the logical unit #13; operating information is

written on the standard output unit and the comment to operator unit.

For an edit operation the MCS control record (card) might appear:

$^{7}_{9}$ELT,MGH,34000,O/13,5,1000.

During operation of the program, a listing is produced on the standard output unit. This listing contains each control card as it appears in the edit deck, all directory cards, and a message PREV REC X for each record transferred from the existing library to the new tape. The completion of the operation is indicated by the message EDIT COMPLETED. A sample edit listing is shown is figure A.1. Error diagnostics are written on the standard output unit and the comment to operator medium as errors are detected. These diagnostics are explained in table 1.

```
START MONITOR RUN.
BEGIN JOB 234
ELT,222,MGH,O/13,5,1000.
                  START
                  1R
PREV.           2
                  3R
      DIRECTRY
      RESIDENT
      0000000000000000
      0430007700014557
      READ*
      WRITE*
      SELECT*
         .
         .
         .
      1205
      1212
      655
         .
         .
         .
      TYPEI/O
      0000000000000000
      0010000000300277
      TYPEI/O
      0
      CBF1*
      CBF2*
      SELECT*
      CARDREAD
      0000000000000001
      0010000000200466
         .
         .
         .
      WRITE*
      LOADER*
      7777777777777777
PREV. TYPEI/O    1
      CARDREAD   1R
PREV. PTI        1
PREV. PTLOG      1
PREV. BFT        1
         .
         .
         .
PREV. MAP        1
PREV. LIBEDIT    1
PREV. COPY*      1
EDIT COMPLETED
```

Figure A.1
Sample Edit Listing

TABLE 1 LIBRARY EDIT ERROR DIAGNOSTICS

CKSUM ERROR XXXXXXXX LTN-50

An input column binary card checksum detected to be in error. XXXXXXXX will be the record number if the error is detected during a file 1 edit; or it will be the name of the subroutine if the error is detected during a file 2 edit.

DIRECTRY ERROR

An error has occurred in the directory.

DIROFLO ERROR XXXXXXXX

One of the eight directory tables requires more than 1/8 of available core. XXXXXXXX will be the name of the routine.

EDITINP ERROR XXXXXXXX

The order of the attempted edit is incorrect or incompatible with the directory. XXXXXXXX will be the name of the routine.

END OF TAPE

An end of tape mark is encountered during writing of the new library tape.

EPT ERROR

Two routines in the directory have the same entry point names.

OFLCORE ERROR XXXXXXXX

Available core is not large enough to allow forming a record. XXXXXXXX will be the record number if the error occurs during a file 1 edit; or it will be the name of the routine if the error occurs during a file 2 edit.

RANGE ERROR XXXXXXXX

An inconsistency exists between the IDC and RBD cards of a file 2 binary deck. XXXXXXXX will be the name of the routine.

READING ERROR XXXXXXXX LTN-YY

A. A persistent (not cleared after three tries) parity error or illegal BCD character is encountered when reading the library tape or the standard input unit.

B. An illegal or impossible number is detected on an edit control card. XXXXXXXX will be the record number if the error is detected during a file 1 edit; or it will be the name of the routine if the error is detected during a file 2 edit.

YY will be the number of the logical unit number on which the error was detected. 00 for the library or 50 for the standard input unit.

WRITING ERROR XXXXXXXX LTN-13

A persistent (not cleared after three tries) parity error is encountered when writing the new library tape. XXXXXXXX will be the number of the record if the error is detected during a file 1 edit; or it will be the name of the routine if the error is detected during a file 2 edit.

```
                                IDENT     GRAFPLOT

RANGE                 FWA   -   LWA+1
                    00000       02123
ENTRY POINTS
                    00022   GRAFPLOT
                    00000   AUTOGRAF
EXTERNAL SYMBOLS
                    00001     FLOATF
                    00002       XTOI
                    00003      XINTF
                    00004      ERROR
                    00005     BUFOUT
                    00006     INGOUT
                    00007       GOUT
                    00010    ENDGOUT
                    00011      WREOF
                    00012     IFUNIT
                    00013        BKSP
                    00014     WRITE*
                    00015     ERROR*
```

```
                    GRAFPLOT - AUTOGRAF
FORTRAN COMPATIBLE SUBROUTINE FOR PLOTTING DATA
      R.L. HOGG / D.C. GLOVER  1 APRIL 1963
            J.S. NAVAL POSTGRADUATE SCHOOL
                 MONTEREY, CALIFORNIA

            INPUT PARAMETERS FOR GRAFPLOT ENTRY

1. MODCURV = 0  FIRST AND LAST CURVE FOR THIS GRA
             1  FIRST CURVE OF MANY
             2  INERMEDIATE CURVE
             3  FINAL CURVE FOR THIS GRAPH

2. NUMPTS  =    NUMBER OF POINTS-MUST BE LESS OR
3. X       =    LOCATION OF X ARRAY
4. Y       =    LOCATION OF Y ARRAY
5. LABEL   =    4 CHARACTER BCD LABEL FOR THIS CU
           =-0  WILL CAUSE AUTOMATIC SEQ. UP TO 8

6. INTERPS = 0  NO INTERPOLATION BETWEEN INPUT PO
           = 1 THRU 7-INTERPOLATE 1 TO 7 INTERMED

7. SFX     = 0  PROGRAM WILL FIND BEST SCALE FACT
           = A  SCALE FACTOR FOR X, UNITS/INCH

8. SFY     = 0  PROGRAM WILL FIND BEST SCALE FACT
           = B  SCALE FACTOR FOR Y, UNITS/INCH

9. IXOFFS  =    NO. OF INCHES TO OFFSET X AXIS
```

```
10.IYOFFS   =     NO. OF INCHES TO OFFSET Y AXIS
11.IWIDE    =     NO. OF INCHES FOR X AXIS, MAX=9
12.IHIGH    =     NO. OF INCHES FOR Y AXIS, MAX=24
13.MODE     = 0   AUTO CAL. BEST OFFSET OF AXES
              1   LOCATE AXES CENTER
              2   LOCATE AXES LOWER LEFT
              3   LOCATE AXES UPPER LEFT
              4   LOCATE AXES UPPER RIGHT
              5   LOCATE AXES LOWER RIGHT

14.TITLESIZ= 1-77           FIXED POINT MULT. FOR
                            LETTER SIZE. BASIC SIZE
                            IS .1 INCHES

15.TITLE1   =     ADDRESS OF 1ST TITLE ARRAY
16.TITLE2   =     ADDRESS OF 2ND TITLE ARRAY
     *            0 TO 2048 TITLES MAY BE USED
     *


                INPUT PARAMETERS FOR AUTOGRAF ENTRY

1. MODCURV
2. NUMPTS
3. X
4. Y
5. TITLE1         0 TO 2048 TITLES MAY BE USED
     *
     *
```

1

```
                00022+          ENTRY     GRAFPLOT
                00000+          ENTRY     AUTOGRAF
00001                           EXT       FLOATF
00002                           EXT       XTOI
00003                           EXT       XINTF
00004                           EXT       ERROR
00005                           EXT       BUFOUT
00006                           EXT       INGOUT
00007                           EXT       GOUT
00010                           EXT       ENDGOUT
00011                           EXT       WREOF
00012                           EXT       IFUNIT
00013                           EXT       BKSP
00014                           EXT       WRITE*
00015                           EXT       ERROR*
```

```
00000+  00  0  77777   AUTOGRAF  ZRO        **
        20  0  01110+             STA        MODCURV
00001+  21  0  01111+             STQ        NUMPTS
        12  0  00000+             LDA        AUTOGRAF
00002+  01  0  00030               ARS        24
        60  0  00004+             SAU        *+2
00003+  11  0  00001               INA        +1
        61  0  00016+             SAL        ARGR2
00004+  12  0  77777   +          LDA        **
        20  0  00007+             STA        ARGU1
00005+  12  0  01110+             LDA        MODCURV
        16  0  01111+             LDQ        NUMPTS
00006+  75  4  00022+ +           SLJ     4  GRAFPLOT
        50  0  00000
1 00007+  00  0  00000   ARGU1     OCT        0
        00  0  00000
00010+  00  0  00015+             ZRO        ALL7S
        00  0  01014+             ZRO        =0
00011+  00  0  01014+             ZRO        =0
        00  0  01014+             ZRO        =0
00012+  00  0  01014+             ZRO        =0
        00  0  01014+             ZRO        =0
00013+  00  0  01022+             ZRO        =9
        00  0  01022+             ZRO        =9
00014+  00  0  01014+             ZRO        =0
        00  0  01017+             ZRO        =2
00015+  77  7  77777   ALL7S     OCT        -0
        77  7  77777
00016+  00  0  00000   ARGR2     OCT        0
        00  0  00000
00017+  50  1  77777   PLOTEXIT  ENI     1  **
        50  2  77777             ENI     2  **
00020+  50  3  77777             ENI     3  **
        50  4  77777             ENI     4  **
00021+  50  5  77777             ENI     5  **
        50  6  77777             ENI     6  **

00022+  75  0  77777   GRAFPLOT  SLJ        **
        50  0  00000             NOP
00023+  56  1  00017+             SIU     1  PLOTEXIT
        57  2  00017+             SIL     2  PLOTEXIT
00024+  56  3  00020+             SIU     3  PLOTEXIT+1
        57  4  00020+             SIL     4  PLOTEXIT+1
00025+  56  5  00021+             SIU     5  PLOTEXIT+2
        57  6  00021+             SIL     6  PLOTEXIT+2
00026+  60  0  00027+             SAU        *+1
        50  0  00000
```

```
00027+  12  0  77777   +          LDA       **
        20  0  01110+              STA       MODCURV
00030+  07  0  00060               LLS       48
        60  0  00031+              SAU       *+1
00031+  12  0  77777   +          LDA       **
        50  0  00000
00032+  20  0  01111+  +          STA       NUMPTS
        11  0  77776               INA       -1
00033+  60  0  00401+              SAU       ST1006
        50  0  00000
00034+  15  0  01035+  +          SUB       =900
        22  3  00036+              AJP    M  *+2
00035+  12  0  01035+  +          LDA       =900
        75  0  00032+              SLJ       *-3
00036+  52  1  00022+  +          LIU    1  GRAFPLOT
        12  1  00000               LDA    1  0
00037+  61  0  01113+              SAL       Y1ADDR
        60  0  00355+              SAU       JPTPROC
00040+  05  0  00030               ALS       24
        61  0  01112+              SAL       X1ADDR
00041+  60  0  00344+              SAU       IPTPROC
        51  1  00001               INI    1  1
1       00042+  12  1  00000       LDA    1  0
        60  0  00043+              SAU       *+1
00043+  12  0  77777   +          LDA       **
        20  0  01121+              STA       INTERPS
00044+  12  1  00000               LDA    1  0
        05  0  00030               ALS       24
00045+  60  0  00046+              SAU       *+1
        50  0  00000
00046+  12  0  77777   +          LDA       **
        20  0  01155+              STA       LABEL
00047+  22  1  00056+              AJP    N  LABJUMP
        22  2  00056+              AJP    P  LABJUMP
00050+  12  0  01110+              LDA       MODCURV
        50  0  00000
00051+  11  0  77775   +          INA       -2
        22  2  00053+              AJP    P  *+2
00052+  12  0  01042+              LDA       LABELCEL
        20  0  01043+              STA       LABELCYC
00053+  10  0  00000               ENA       0
        16  0  01043+              LDQ       LABELCYC
00054+  07  0  00006               LLS       6
        21  0  01043+              STQ       LABELCYC
00055+  05  0  00044               ALS       36
        20  0  01155+              STA       LABEL
00056+  12  0  01110+  LABJUMP     LDA       MODCURV
        11  0  77775               INA       -2
00057+  22  3  00063+              AJP    M  SET2SCAL
```

```
                51  1  00001              INI     1  1
00060+          57  1  01120+             SIL     1  T1/2ADDR
                51  1  00004              INI     1  4
00061+          12  1  00000              LDA     1  0
                22  2  00124+             AJP     P  HOWMANY
00062+          22  1  00124+             AJP     N  HOWMANY
                75  0  00106+             SLJ        SET4AUTO
00063+          51  1  00001   SET2SCAL    INI     1  1
                12  1  00000              LDA     1  0
00064+          61  0  01115+             SAL        SFYADDR
                05  0  00030              ALS        24
00065+          61  0  01114+             SAL        SFXADDR
                51  1  00001              INI     1  1
00066+          12  1  00000              LDA     1  0
                60  0  00071+             SAU        *+3
00067+          05  0  00030              ALS        24
                60  0  00070+             SAU        *+1
00070+          12  0  77777   +          LDA        **
                20  0  01075+             STA        IXOFFS
00071+          12  0  77777   +          LDA        **
                20  0  01076+             STA        IYOFFS
00072+          51  1  00001              INI     1  1
                12  1  00000              LDA     1  0
00073+          60  0  00076+             SAU        *+3
                05  0  00030              ALS        24
00074+          60  0  00075+             SAU        *+1
                50  0  00000
00075+          12  0  77777   +          LDA        **
                20  0  01077+             STA        IWIDE
1       00076+  12  0  77777   +          LDA        **
                20  0  01115+             STA        IHIGH
00077+          51  1  00001              INI     1  1
                12  1  00000              LDA     1  0
00100+          60  0  00102+             SAU        *+2
                05  0  00030              ALS        24
00101+          61  0  01067+             SAL        MODEADDR
                50  0  00000
00102+          12  0  77777   +          LDA        **
                20  0  01117+             STA        TITLESIZ
00103+          51  1  00001              INI     1  1
                57  1  01120+             SIL     1  T1/2ADDR
00104+          12  1  00000   +          LDA     1  0
                22  2  00110+             AJP     P  *+4
00105+          22  1  00110+             AJP     N  *+3
                50  0  00000
00106+          51  1  00001   SET4AUTO    INI     1  1
                12  1  00000              LDA     1  0
00107+          41  0  01050+             SCL        CLEARMK
                20  0  01120+             STA        T1/2ADDR
```

```
00110+  12  0  01077+          LDA      IWIDE
        15  0  01022+          SUB      =9
00111+  22  0  00113+  +       AJP   Z  *+2
        22  3  00113+          AJP   M  *+2
00112+  12  0  01022+          LDA      =9
        20  0  01077+          STA      IWIDE
00113+  12  0  01115+          LDA      IHIGH
        15  0  01025+          SUB      =20
00114+  22  0  00115+  +       AJP   Z  *+2
        22  3  00115+          AJP   M  *+2
00115+  12  0  01025+          LDA      =20
        20  0  01115+          STA      IHIGH
00116+  10  0  01077+          ENA      IWIDE
        75  4  X00001          RTJ      FLOATF
00117+  75  4  X00004   +       RTJ      ERROR
        50  0  00000
00120+  20  0  01064+  +       STA      WIDTH
        10  0  01115+          ENA      IHIGH
00121+  75  4  X00001          RTJ      FLOATF
        50  0  00000
00122+  75  4  X00004   +       RTJ      ERROR
        50  0  00000
00123+  20  0  01065+  +       STA      HEIGHT
        50  1  00000          ENI   1  0
00124+  12  0  01120+  HOWMANY  LDA      T1/2ADDR
        20  0  01170+          STA      ITEMP
00125+  60  0  00125+          SAU      *+1
        50  0  00000
00126+  12  0  77777   +       LDA      **
        20  0  01171+          STA      ITEMP+1
00127+  16  0  01047+          LDQ      LOWMASK
        44  0  01171+          LDL      ITEMP+1
00130+  22  1  00132+  +       AJP   N  *+2
        51  1  00002          INI   1  2
00131+  72  0  01170+          RAO      ITEMP
        75  0  00125+          SLJ      HOWMANY+1
1       00132+  06  0  00030          QLS      24
        44  0  01171+          LDL      ITEMP+1
00133+  22  1  00135+  +       AJP   N  *+2
        72  0  01170+          RAO      ITEMP
00134+  51  1  00001          INI   1  1
        50  0  00000
00135+  57  1  01154+  +       SIL   1  KTITLE
        12  0  01170+          LDA      ITEMP
00136+  11  0  00001          INA      1
        60  0  00022+          SAU      GRAFPLOT

                               TEST IF FIRST CURVE ENTRY, THIS GRAPH
```

```
00137+  12  0  01110+  +          LDA       MODCURV
        11  0  77775               INA       -2
00140+  22  2  00340+              AJP     P PLOTCURV          NOT 1ST, GO PLOT CURVE
        50  0  00000
00141+  75  4  00435+  +          RTJ       SCALE             GO TEST/PERFORM SCALING
        50  0  00000
00142+  12  0  01102+  CALCVALS   LDA       XSCALE
        20  0  01072+              STA       X/YSCALE
00143+  75  4  00717+              RTJ       ENCODE
        50  0  00000
00144+  12  0  01170+  +          LDA       ITEMP
        20  0  01104+              STA       XSCLAB1
00145+  12  0  01171+              LDA       ITEMP+1
        20  0  01105+              STA       XSCLAB2
00146+  16  0  01044+              LDQ       MK1
        44  0  01170+              LDL       ITEMP
00147+  05  0  00014               ALS       12
        20  0  01100+              STA       LABELX
00150+  12  0  01103+              LDA       YSCALE
        20  0  01072+              STA       X/YSCALE
00151+  75  4  00717+              RTJ       ENCODE
        50  0  00000
00152+  12  0  01170+  +          LDA       ITEMP
        20  0  01106+              STA       YSCLAB1
00153+  12  0  01171+              LDA       ITEMP+1
        20  0  01107+              STA       YSCLAB2
00154+  16  0  01044+              LDQ       MK1
        44  0  01170+              LDL       ITEMP
00155+  05  0  00014               ALS       12
        20  0  01101+              STA       LABELY
00156+  10  0  00000               ENA       0
        20  0  01136+              STA       IH
00157+  20  0  01137+              STA       JV
        12  0  01075+              LDA       IXOFFS
00160+  24  0  01033+              MUI       =100
        20  0  01140+              STA       JH
00161+  15  0  01026+              SUB       =25
        20  0  01141+              STA       JHL
00162+  12  0  01077+              LDA       IWIDE
        24  0  01033+              MUI       =100
00163+  20  0  01142+              STA       LH
        14  0  01136+              ADD       IH
00164+  15  0  01021+              SUB       =8
        20  0  01143+              STA       IHL
1       00165+  13  0  01033+      LAC       =100
        20  0  01144+              STA       INCRIHL
00166+  20  0  01122+              STA       INCRJVL
        12  0  01077+              LDA       IWIDE
00167+  15  0  01076+              SUB       IYOFFS
```

```
                24   0   01100+          MUI         LABELX
00170+          20   0   01123+          STA         IVH
                13   0   01100+          LAC         LABELX
00171+          20   0   01124+          STA         INCRVH
                12   0   01077+          LDA         IWIDE
00172+          11   0   00001           INA         1
                20   0   01125+          STA         NH
00173+          10   0   00204           ENA         2048
                20   0   01150+          STA         LSHOH
00174+          12   0   01076+          LDA         IYOFFS
                24   0   01033+          MUI         =100
00175+          20   0   01145+          STA         IV
                12   0   01116+          LDA         IHIGH
00176+          24   0   01033+          MUI         =100
                20   0   01146+          STA         LV
00177+          12   0   01076+          LDA         IYOFFS
                24   0   01033+          MUI         =100
00200+          15   0   01031+          SUB         =70
                20   0   01126+          STA         IVL
00201+          12   0   01146+          LDA         LV
                11   0   00012           INA         10
00202+          20   0   01127+          STA         JVL
                12   0   01116+          LDA         IHIGH
00203+          15   0   01075+          SUB         IXOFFS
                24   0   01101+          MUI         LABELY
00204+          20   0   01130+          STA         IVV
                13   0   01101+          LAC         LABELY
00205+          20   0   01131+          STA         INCRVV
                12   0   01116+          LDA         IHIGH
00206+          11   0   00001           INA         1
                20   0   01132+          STA         NV
00207+          10   0   00204           ENA         2048
                20   0   01133+          STA         LSVOV
00210+          12   0   01136+          LDA         IH
                20   0   01134+          STA         IT
00211+          13   0   01053+          LAC         LOWMARG
                24   0   01033+          MUI         =100
00212+          20   0   01135+          STA         JT
                10   0   00004           ENA         4
00213+          20   0   01147+          STA         ITOR
                12   0   01154+          LDA         KTITLE
00214+          11   0   00002           INA         2
                20   0   01154+          STA         KTITLE

                                                    WRITE TITLE CONTROL RECORD

00215+          16   0   01045+          LDQ         MK2
                44   0   01134+          LDL         IT
00216+          05   0   00014           ALS         12
```

```
                43  0  01135+          SSU       JT
          00217+ 05  0  00014          ALS       12
                43  0  01147+          SSU       ITOR
1         00220+ 05  0  00014          ALS       12
                43  0  01154+          SSU       KTITLE
          00221+ 20  0  01170+          STA       ITEMP
                50  0  00000
          00222+ 10  0  01015+ +       ENA       =1
                04  0  01027+          ENQ       =48
          00223+ 75  4  X00005         RTJ       BUFOUT
                50  0  00000
          00224+ 00  0  01170+ +       ZRO       ITEMP
                00  0  01170+          ZRO       ITEMP
          00225+ 75  4  X00004   +     RTJ       ERROR
                50  0  00000
          00226+ 75  4  00737+ +       RTJ       TPCHK
                50  0  00000

                                       WRITE OUT AXIS SCALE TITLES

          00227+ 12  0  01056+ +       LDA       SCALEX1
                16  0  01044+          LDQ       MK1
          00230+ 06  0  00006          QLS       6
                43  0  01054+          SSU       SCALSIZ
          00231+ 20  0  01170+          STA       ITEMP
                12  0  01057+          LDA       SCALEX2
          00232+ 20  0  01171+          STA       ITEMP+1
                12  0  01104+          LDA       XSCLAB1
          00233+ 20  0  01172+          STA       ITEMP+2
                12  0  01105+          LDA       XSCLAB2
          00234+ 20  0  01173+          STA       ITEMP+3
                50  0  00000
          00235+ 10  0  01015+ +       ENA       =1
                04  0  01027+          ENQ       =48
          00236+ 75  4  X00005         RTJ       BUFOUT
                50  0  00000
          00237+ 00  0  01170+ +       ZRO       ITEMP
                00  0  01173+          ZRO       ITEMP+3
          00240+ 75  4  X00004   +     RTJ       ERROR
                50  0  00000
          00241+ 75  4  00737+ +       RTJ       TPCHK
                50  0  00000
          00242+ 12  0  01060+ +       LDA       SCALEY1
                16  0  01044+          LDQ       MK1
          00243+ 06  0  00006          QLS       6
                43  0  01054+          SSU       SCALSIZ
          00244+ 20  0  01202+          STA       ITEMP+10
                12  0  01061+          LDA       SCALEY2
          00245+ 20  0  01203+          STA       ITEMP+11
```

```
                 12  0  01105+                LDA       YSCLAB1
00246+           20  0  01204+                STA       ITEMP+12
                 12  0  01107+                LDA       YSCLAB2
00247+           20  0  01205+                STA       ITEMP+13
                 50  0  00000
00250+           10  0  01015+  +             ENA       =1
                 04  0  01027+                ENQ       =48
00251+           75  4  X00005                RTJ       BUFOUT
                 50  0  00000
00252+           00  0  01202+  +             ZRO       ITEMP+10
                 00  0  01205+                ZRO       ITEMP+13
1    00253+      75  4  X00004    +           RTJ       ERROR
                 50  0  00000
00254+           75  4  00737+  +             RTJ       TPCHK
                 50  0  00000

                                                       LOOP TO WRITE OUT TITLES
00255+           53  1  01154+                LIL    1  KTITLE
                 51  1  77775                 INI    1  -2
00256+           12  0  01051+                LDA       CONTROL2
                 20  0  01052+                STA       CONTR1
00257+           12  0  01120+  SETITLE1      LDA       T1/2ADDR
                 60  0  00260+                SAU       *+1
00260+           12  0  77777    +           LDA       **
                 05  0  00030                ALS       24
00261+           60  0  00265+  SETITLE2      SAU       STRTITLE
                 50  0  00000
00262+           55  1  00263+  +             IJP    1  *+1
                 75  0  00306+                SLJ       CONTROL
00263+           50  4  00024                 ENI    4  20
                 10  0  00000                 ENA       0
00264+           20  4  01207+  KLEER         STA    4  STORE
                 55  4  00264+                IJP    4  KLEER
00265+           50  2  00000    +           ENI    2  0
                 12  0  01117+                LDA       TITLESIZ
00266+           16  2  77777    STRTITLE     LDQ    2  **
                 07  0  00052                 LLS       42
00267+           20  2  01207+                STA    2  STORE
                 07  0  00006                 LLS       6
00270+           20  0  01225+                STA       STORE+14
                 50  0  00000
00271+           12  0  01117+  +             LDA       TITLESIZ
                 24  0  01021+                MUI       =8
00272+           70  0  01225+                RAD       STORE+15
                 15  0  01033+                SUB       =100
00273+           22  2  00275+                AJP    P  NOMORR
                 12  0  01225+                LDA       STORE+14
00274+           54  2  00011    +           ISK    2  9
```

```
                75   0   00266+                SLJ         STRTITLE
00275+          10   0   01015+  NOMORR        ENA         =1
                04   0   01027+                ENQ         =48
00276+          75   4  X00005                 RTJ         BUFOUT
                50   0   00000
00277+          00   0   01207+  +             ZRO         STORE
                00   0   01221+                ZRO         STORE+10
00300+          75   4  X00004    +            RTJ         ERROR
                50   0   00000
00301+          75   4   00737+  +             RTJ         TPCHK
                50   0   00000
00302+          37   0   01052+  +             SSH         CONTR1
                75   0   00305+                SLJ         *+3
00303+          12   0   01120+  +             LDA         T1/2ADDR
                60   0   00304+                SAU         *+1
00304+          12   0   77777    +            LDA         **
                75   0   00261+                SLJ         SETITLE2
00305+          72   0   01120+  +             RAO         T1/2ADDR
                75   0   00257+                SLJ         SETITLE1
1
                                                           WRITE AXIS  CONTROL RECORD
00306+          16   0   01045+  CONTROL       LDQ         MK2
                44   0   01136+                LDL         IH
00307+          05   0   00014                 ALS         12
                43   0   01140+                SSU         JH
00310+          05   0   00014                 ALS         12
                43   0   01142+                SSU         LH
00311+          05   0   00014                 ALS         12
                43   0   01143+                SSU         IHL
00312+          20   0   01170+                STA         ITEMP
                44   0   01141+                LDL         JHL
00313+          05   0   00014                 ALS         12
                43   0   01144+                SSU         INCRIHL
00314+          05   0   00014                 ALS         12
                43   0   01123+                SSU         IVH
00315+          05   0   00014                 ALS         12
                43   0   01124+                SSU         INCRVH
00316+          20   0   01171+                STA         ITEMP+1
                44   0   01125+                LDL         NH
00317+          05   0   00014                 ALS         12
                43   0   01150+                SSU         LSHOH
00320+          05   0   00014                 ALS         12
                43   0   01145+                SSU         IV
00321+          05   0   00014                 ALS         12
                43   0   01137+                SSU         JV
00322+          20   0   01172+                STA         ITEMP+2
                44   0   01146+                LDL         LV
00323+          05   0   00014                 ALS         12
```

```
                43   0   01126+                SSU         IVL
00324+          05   0   00014                 ALS         12
                43   0   01127+                SSU         JVL
00325+          05   0   00014                 ALS         12
                43   0   01122+                SSU         INCRJVL
00326+          20   0   01173+                STA         ITEMP+3
                44   0   01130+                LDL         IVV
00327+          05   0   00014                 ALS         12
                43   0   01131+                SSU         INCRVV
00330+          05   0   00014                 ALS         12
                43   0   01132+                SSU         NV
00331+          05   0   00014                 ALS         12
                43   0   01133+                SSU         LSVOV
00332+          20   0   01174+                STA         ITEMP+4
                50   0   00000
00333+          10   0   01015+  +             ENA         =1
                04   0   01027+                ENQ         =48
00334+          75   4  X00005                 RTJ         BUFOUT
                50   0   00000
00335+          00   0   01170+  +             ZRO         ITEMP
                00   0   01174+                ZRO         ITEMP+4
00336+          75   4  X00004    +            RTJ         ERROR
                50   0   00000
00337+          75   4   00737+  +             RTJ         TPCHK
                50   0   00000

1                                                         WRITE CURVE DATA RECORD

00340+          10   0   01210+ PLOTCURV       ENA         STORE+1
                41   0   01050+                SCL         CLEARMK
00341+          61   0   01153+                SAL         BUFAD
                12   0   01051+                LDA         CONTROL2
00342+          20   0   01052+                STA         CONTR1
                12   0   01121+                LDA         INTERPS
00343+          20   0   01207+                STA         STORE
                50   3   00000                 ENI     3   0
00344+          12   3   77777   IPTPROC       LDA     3   **
                32   0   01034+                FMU         =100.0
00345+          33   0   01102+                FDV         XSCALE
                20   0   01170+                STA         ITEMP
00346+          10   0   01170+                ENA         ITEMP
                75   4  X00003                 RTJ         XINTF
00347+          75   4  X00004    +            RTJ         ERROR
                50   0   00000
00350+          14   0   01145+  +             ADD         IV
                50   0   00000
00351+          22   2   00352+  +             AJP     P   *+1
                10   0   00000                 ENA         0
00352+          20   0   01151+                STA         IPT
```

```
                15   0   01142+              SUB       LH
00353+          22   3   00355+              AJP     M JPTPROC
                12   0   01142+              LDA       LH
00354+          20   0   01151+              STA       IPT
                50   0   00000
00355+          12   3   77777   JPTPROC     LDA     3 **
                32   0   01034+              FMU       =100.0
00356+          33   0   01103+              FDV       YSCALE
                20   0   01170+              STA       ITEMP
00357+          10   0   01170+              ENA       ITEMP
                75   4  X00003              RTJ       XINTF
00360+          75   4  X00004   +          RTJ       ERROR
                50   0   00000
00361+          14   0   01140+  +           ADD       JH
                50   0   00000
00362+          22   2   00363+  +           AJP     P *+1
                10   0   00000              ENA       0
00363+          20   0   01152+              STA       JPT
                15   0   01146+              SUB       LV
00364+          22   3   00366+              AJP     M STORENTR
                12   0   01146+              LDA       LV
00365+          20   0   01152+              STA       JPT
                50   0   00000
00366+          04   0   07777   STORENTR    ENQ       7777B
                44   0   01151+              LDL       IPT
00367+          37   0   01052+              SSH       CONTR1
                75   0   00374+              SLJ       ST1005
00370+          05   0   00044              ALS       36
                20   7   01153+              STA     7 BUFAD
00371+          44   0   01152+              LDL       JPT
                05   0   00030              ALS       24
00372+          06   0   00044              QLS       36
                43   7   01153+              SSU     7 BUFAD
1       00373+  20   7   01153+              STA     7 BUFAD
                75   0   00401+              SLJ       ST1006
00374+          05   0   00014   ST1005      ALS       12
                16   0   01046+              LDQ       MASK
00375+          43   7   01153+              SSU     7 BUFAD
                04   0   07777              ENQ       7777B
00376+          43   0   01152+              SSU       JPT
                20   7   01153+              STA     7 BUFAD
00377+          72   0   01153+              RAO       BUFAD
                10   0   00000              ENA       0
00400+          20   7   01153+              STA     7 BUFAD
                50   0   00000
00401+          54   3   77777   ST1006      ISK     3 **
                75   0   00344+              SLJ       IPTPROC
00402+          12   7   01153+              LDA     7 BUFAD
                04   0   00000              ENQ       0
```

```
00403+  07  0  00030          LLS     24
        23  0  00414+         QJP     ST1007
00404+  07  0  00060          LLS     48
        16  0  01036+         LDQ     NFLAG
00405+  07  0  00030          LLS     24
        20  7  01153+         STA   7 BUFAD
00406+  72  0  01153+         RAO     BUFAD
        50  0  00000          NOP
00407+  10  0  00000    +     ENA     0
        16  0  01155+         LDQ     LABEL
00410+  07  0  00030          LLS     24
        04  0  00104          ENQ     104B
00411+  06  0  00044          QLS     36
        07  0  00030          LLS     24
00412+  20  7  01153+         STA   7 BUFAD
        12  0  01153+         LDA     BUFAD
00413+  61  0  00423+         SAL     INITERM
        75  0  00421+         SLJ     ST1008
00414+  12  0  01037+ ST1007   LDA     INFLAG
        16  0  01155+         LDQ     LABEL
00415+  07  0  00030          LLS     24
        20  7  01153+         STA   7 BUFAD
00416+  72  0  01153+         RAO     BUFAD
        10  0  00104          ENA     104B
00417+  05  0  00044          ALS     36
        20  7  01153+         STA   7 BUFAD
00420+  12  0  01153+         LDA     BUFAD
        61  0  00423+         SAL     INITERM
00421+  10  0  01015+ ST1008   ENA     =1
        04  0  01027+         ENQ     =48
00422+  75  4  X00005          RTJ     BUFOUT
        50  0  00000
00423+  00  0  01207+ INITERM  ZRO     STORE
        00  0  77777          ZRO     **
00424+  75  4  X00004    +     RTJ     ERROR
        50  0  00000
00425+  75  4  00737+   +     RTJ     TPCHK
        50  0  00000
00426+  12  0  01110+   +     LDA     MODCURV
        22  3  00431+         AJP   M *+3
1       00427+  22  0  00431+         AJP   Z *+2
        50  0  00000
00430+  11  0  77774    +     INA     -3
        22  3  00434+         AJP   M *+4
00431+  10  0  01027+   +     ENA     =48
        75  4  X00011          RTJ     WREOF
00432+  75  4  X00004    +     RTJ     ERROR
        50  0  00000
00433+  12  0  01042+   +     LDA     LABELCEL
```

```
                76  1  00434+          SLS     1  *+1
00434+          75  0  00017+  +       SLJ        PLOTEXIT
                50  0  00000
00435+          75  0  77777    SCALE   SLJ        **
                12  0  01114+          LDA        SFXADDR
00436+          60  0  00437+          SAU        *+1
                50  0  00000
00437+          12  0  77777    +       LDA        **
                20  0  01102+          STA        XSCALE
00440+          22  1  00443+  +       AJP     N  *+3
                75  4  00451+          RTJ        SCALEX
00441+          10  0  01102+  +       ENA        XSCALE
                04  0  01076+          ENQ        IYOFFS
00442+          75  4  00732+  +       RTJ        ALLZEROS
                50  0  00000
00443+          12  0  01115+  +       LDA        SFYADDR
                60  0  00444+          SAU        *+1
00444+          12  0  77777    +       LDA        **
                20  0  01103+          STA        YSCALE
00445+          22  1  00450+  +       AJP     N  *+3
                75  4  00475+          RTJ        SCALEY
00446+          10  0  01103+  +       ENA        YSCALE
                04  0  01075+          ENQ        IXOFFS
00447+          75  4  00732+  +       RTJ        ALLZEROS
                50  0  00000
00450+          75  0  00435+  +       SLJ        SCALE
                50  0  00000
00451+          75  0  77777    SCALEX  SLJ        **
                12  0  01112+          LDA        X1ADDR
00452+          16  0  01111+          LDQ        NUMPTS
                75  4  00577+          RTJ        MAX/MIN
00453+          20  0  01155+  +       STA        XMAX
                21  0  01161+          STQ        XMIN
00454+          22  2  00455+  +       AJP     P  *+1          SET ABSOLUTE
                10  0  00000          ENA        0                  XMAX OR
00455+          20  0  01157+          STA        AXMAX                 (=0 IF XMAX IS -)
                20  0  01160+          STA        AMAX
00456+          12  0  01161+          LDA        XMIN            SET
                50  0  00000
00457+          22  3  00460+  +       AJP     M  *+1                ABSOLUTE
                10  0  77777          ENA        -0                    XMIN OR
00460+          42  0  00015+          SCM        ALL7S                    (=0 IF XMIN IS +)
                20  0  01162+          STA        AXMIN
00461+          20  0  01163+          STA        AMIN
                12  0  01064+          LDA        WIDTH           GO SET
00462+          20  0  01063+          STA        LENGTH             AXMAX/AXMIN/XTOTAL
                10  0  77777          ENA        -0                    IN ACCORDANCE
1               00463+  20  0  01040+          STA        X/YFLAG               WITH MODE SELECTED
                75  4  00652+          RTJ        GETTOTAL
```

```
00464+  12  0  01062+ +        LDA      TOTAL
        22  1  00466+           AJP   N  *+2
00465+  75  0  00451+           SLJ      SCALEX
        50  0  00000
00466+  52  1  00451+ +        LIU   1  SCALEX          BUILD
        51  1  00002           INI   1  2                  NORMAL
00467+  56  1  00451+           SIU   1  SCALEX                RETURN
        75  4  00521+           RTJ      SCALEX/Y     GO SCALE X
00470+  12  0  01072+ +        LDA      X/YSCALE
        20  0  01102+           STA      XSCALE
00471+  12  0  01063+           LDA      LENGTH
        20  0  01064+           STA      WIDTH
00472+  75  4  00622+           RTJ      CALCOFFS
        50  0  00000
00473+  12  0  01074+ +        LDA      IOFFS
        20  0  01076+           STA      IYOFFS
00474+  75  0  00451+           SLJ      SCALEX
        50  0  00000
00475+  75  0  77777   SCALEY   SLJ      **
        12  0  01113+           LDA      Y1ADDR
00476+  16  0  01111+           LDQ      NUMPTS
        75  4  00577+           RTJ      MAX/MIN
00477+  20  0  01164+ +        STA      YMAX
        21  0  01165+           STQ      YMIN
00500+  22  2  00501+ +        AJP   P  *+1              SET ABSOLUTE
        10  0  00000           ENA      0                    YMAX OR
00501+  20  0  01165+           STA      AYMAX                     (=0 IF YMAX IS -)
        20  0  01160+           STA      AMAX
00502+  12  0  01165+           LDA      YMIN             SET
        50  0  00000
00503+  22  3  00504+ +        AJP   M  *+1                 ABSOLUTE
        10  0  77777           ENA      -0                   YMIN OR
00504+  42  0  00015+           SCM      ALL7S                     (=0 IF YMIN IS +)
        20  0  01167+           STA      AYMIN
00505+  20  0  01163+           STA      AMIN
        12  0  01065+           LDA      HEIGHT
00506+  20  0  01063+           STA      LENGTH
        10  0  00000           ENA      +0
00507+  20  0  01040+           STA      X/YFLAG          SET FOR Y SCALING
        75  4  00652+           RTJ      GETTOTAL
00510+  12  0  01062+ +        LDA      TOTAL
        22  1  00512+           AJP   N  *+2
00511+  75  0  00475+           SLJ      SCALEY
        50  0  00000
00512+  52  1  00475+ +        LIU   1  SCALEY           BUILD
        51  1  00002           INI   1  2                  NORMAL
00513+  56  1  00475+           SIU   1  SCALEY                RETURN
        75  4  00521+           RTJ      SCALEX/Y     GO SCALE Y
00514+  12  0  01072+ +        LDA      X/YSCALE
```

```
                20  0  01103+            STA       YSCALE
          00515+ 12  0  01063+            LDA       LENGTH
                20  0  01065+            STA       HEIGHT
          00516+ 75  4  00622+            RTJ       CALCOFFS
                50  0  00000
1         00517+ 12  0  01074+ +          LDA       IOFFS
                20  0  01075+            STA       IXOFFS
          00520+ 75  0  00475+            SLJ       SCALEY
                50  0  00000
          00521+ 75  0  77777  SCALEX/Y   SLJ       **                    EXIT
                10  0  00001            ENA       1
          00522+ 20  0  01041+            STA       O/UFLAG
                20  0  01066+            STA       EXPONENT
          00523+ 50  2  00000            ENI     2 0
                12  0  00574+            LDA       QUANTAB
          00524+ 20  0  01070+            STA       QUANTA
                50  0  00000
          00525+ 10  0  01024+ TESTSCAL   ENA       =10.0
                04  0  01065+            ENQ       EXPONENT
          00526+ 75  4  X00002            RTJ       XTOI
                50  0  00000
          00527+ 75  4  X00004 +          RTJ       ERROR
                50  0  00000
          00530+ 32  0  01070+ +          FMU       QUANTA
                20  0  01071+            STA       TEMPSCAL
          00531+ 32  0  01063+            FMU       LENGTH
                31  0  01062+            FSB       TOTAL
          00532+ 22  0  00534+            AJP     Z HITSCALE
                22  3  00536+            AJP     M UNDRTEST
          00533+ 75  0  00542+            SLJ       OVERTEST
                50  0  00000
          00534+ 12  0  01071+ HITSCALE   LDA       TEMPSCAL
                20  0  01072+            STA       X/YSCALE
          00535+ 75  0  00521+            SLJ       SCALEX/Y
                50  0  00000
          00536+ 12  0  01041+ UNDRTEST   LDA       O/UFLAG               RESCALE IF FIRST PASS
                22  1  00550+            AJP     N URESCALE
          00537+ 36  0  01041+ +          SSK       O/UFLAG               O/UFLAG=+ IF OVER BEFORE
                75  0  00554+            SLJ       UP1/OUT                       - IF UNDER BEFORE
          00540+ 75  4  00562+            RTJ       INCQUANT              INCREASE SCALE
                50  0  00000
          00541+ 75  0  00525+ +          SLJ       TESTSCAL
                50  0  00000
          00542+ 12  0  01041+ OVERTEST   LDA       O/UFLAG
                22  1  00552+            AJP     N ORESCALE
          00543+ 36  0  01041+            SSK       O/UFLAG
                75  0  00545+            SLJ       LOWERSCL
          00544+ 12  0  01071+            LDA       TEMPSCAL
                20  0  01072+            STA       X/YSCALE
```

```
00545+  75  0  00521+          SLJ       SCALEX/Y
        50  0  00000
00546+  75  4  00566+ LOWERSCL RTJ       DECQUANT
        50  0  00000
00547+  75  0  00525+ +         SLJ       TESTSCAL
        50  0  00000
00550+  10  0  77777  URESCALE ENA       -0
        20  0  01041+          STA       O/UFLAG
00551+  75  0  00525+          SLJ       TESTSCAL
        50  0  00000
00552+  10  0  00000  ORESCALE ENA       0
        20  0  01041+          STA       O/UFLAG
1 00553+  75  0  00525+          SLJ       TESTSCAL
        50  0  00000
00554+  75  4  00562+ UP1/OUT  RTJ       INCQUANT
        50  0  00000
00555+  10  0  01024+ +         ENA       =10.0
        04  0  01066+          ENQ       EXPONENT
00556+  75  4 X00002           RTJ       XTOI
        50  0  00000
00557+  75  4 X00004  +         RTJ       ERROR
        50  0  00000
00560+  32  0  01070+ +         FMU       QUANTA
        20  0  01072+          STA       X/YSCALE
00561+  75  0  00521+          SLJ       SCALEX/Y
        50  0  00000
00562+  75  0  77777  INCQUANT SLJ       **
        52  3  00562+          LIU    3  *
00563+  56  3  00573+          SIU    3  GOTEST+1
        50  0  00000
00564+  54  2  00002  +         ISK    2  2
        75  0  00572+          SLJ       GOTEST
00565+  72  0  01066+          RAO       EXPONENT
        75  0  00572+          SLJ       GOTEST
00566+  75  0  77777  DECQUANT SLJ       **
        52  3  00566+          LIU    3  *
00567+  56  3  00573+          SIU    3  GOTEST+1
        50  0  00000
00570+  55  2  00572+ +         IJP    2  GOTEST
        50  0  00000
00571+  73  0  01066+ +         RSO       EXPONENT
        50  2  00002           ENI    2  2
00572+  12  2  00574+ GOTEST   LDA    2  QUANTAB
        20  0  01070+          STA       QUANTA
00573+  75  0  77777           SLJ       **
        50  0  00000
00574+  20  0  14000  QUANTAB  DEC       1.0
        00  0  00000
00575+  20  0  24000           DEC       2.0
```

```
                00  0  00000
00576+          20  0  35000                DEC       5.0
                00  0  00000
00577+          75  0  77777   MAX/MIN      SLJ       **
                56  1  00617+               SIU    1  RESTOR
00600+          60  0  00605+               SAU       SET1
                61  0  00607+               SAL       OPERATE
00601+          61  0  00610+               SAL       SET2
                61  0  00612+               SAL       NOTBIGR
00602+          61  0  00613+               SAL       SET3
                07  0  00060                LLS       48                Q TO A
00603+          11  0  77776                INA       -1
                60  0  00615+               SAU       TESTING           SET NUMBER IN ARRAY
00604+          50  1  00000                ENI    1  0
                50  0  00000
00605+          12  1  77777   SET1         LDA    1  **
                20  0  00620+               STA       MAXARG            PRIME ROUTINE
00606+          20  0  00621+               STA       MINARG
                50  0  00000
1   00607+      12  0  00620+  OPERATE      LDA       MAXARG            LOOP
                31  1  77777                FSB    1  **
00610+          22  2  00612+  SET2         AJP    2  NOTBIGR           JUMP IF POSITIVE
                12  1  77777                LDA    1  **
00611+          20  0  00620+               STA       MAXARG            NEW MAXARG
                75  0  00615+               SLJ       TESTING
00612+          12  0  00621+  NOTBIGR      LDA       MINARG
                31  1  77777                FSB    1  **
00613+          22  3  00615+  SET3         AJP    3  TESTING           JUMP IF NEGATIVE
                12  1  77777                LDA    1  **
00614+          20  0  00621+               STA       MINARG            NEW MINARG
                50  0  00000                ENI       0
00615+          54  1  77777   TESTING      ISK    1  **                TEST FOR ALL ARGUMENTS
                75  0  00607+               SLJ       OPERATE
00616+          12  0  00620+               LDA       MAXARG
                16  0  00621+               LDQ       MINARG
00617+          50  1  77777   RESTOR       ENI    1  **
                75  0  00577+               SLJ       MAX/MIN           EXIT
00620+                         MAXARG       BSS       1
00621+                         MINARG       BSS       1

00622+          75  0  77777   CALCOFFS     SLJ       **
                12  7  01067+               LDA    7  MODEADDR
00623+          22  2  00624+               AJP    P  MODE0
                10  0  00000                ENA       0
00624+          22  1  00632+  MODE0        AJP    N  MODE1
                12  0  01163+               LDA       AMIN
00625+          33  0  01072+               FDV       X/YSCALE
                30  0  01015+               FAD       =1.0
00626+          20  0  01073+  CONVERT      STA       TEMP
```

```
          10  0  01073+            ENA       TEMP
00627+    75  4 X00003            RTJ       XINTF
          50  0  00000
00630+    75  4 X00004   +        RTJ       ERROR
          50  0  00000
00631+    20  0  01074+  +         STA       IOFFS
          75  0  00622+            SLJ       CALCOFFS
00632+    11  0  77775   MODE1     INA       -1
          22  1  00635+            AJP     N MODE2
00633+    12  0  01063+            LDA       LENGTH
          33  0  01020+            FDV       =2.0
00634+    75  0  00625+            SLJ       CONVERT
          50  0  00000
00635+    11  0  77775   MODE2     INA       -1
          22  1  00637+            AJP     N MODE3
00636+    10  0  00000            ENA       0
          75  0  00625+            SLJ       CONVERT
00637+    11  0  77775   MODE3     INA       -1
          22  1  00643+            AJP     N MODE4
00640+    36  0  01040+            SSK       X/YFLAG
          75  0  00642+            SLJ       SETMOD3X
00641+    10  0  00000            ENA       0
          75  0  00625+            SLJ       CONVERT
00642+    12  0  01063+ SETMOD3X   LDA       LENGTH
          75  0  00625+            SLJ       CONVERT
00643+    11  0  77775   MODE4     INA       -1
          22  1  00645+            AJP     N MODE5
1 00644+  12  0  01063+            LDA       LENGTH
          75  0  00626+            SLJ       CONVERT
00645+    11  0  77775   MODE5     INA       -1
          22  1  00651+            AJP     N MODERR
00646+    36  0  01040+            SSK       X/YFLAG
          75  0  00650+            SLJ       SETMOD5X
00647+    12  0  01063+            LDA       LENGTH
          75  0  00625+            SLJ       CONVERT
00650+    10  0  00000   SETMOD5X   ENA       0
          75  0  00625+            SLJ       CONVERT
00651+    10  0  00000   MODERR     ENA       0
          75  0  00624+            SLJ       MODE0
00652+    75  0  77777   GETTOTAL   SLJ       **
          12  7  01067+            LDA     7 MODEADDR
00653+    22  2  00654+            AJP     P MODE0CK
          10  0  00000            ENA       0
00654+    22  1  00660+ MODE0CK    AJP     N MODE1CK
          12  0  01160+            LDA       AMAX
00655+    30  0  01163+            FAD       AMIN
          20  0  01062+            STA       TOTAL
00656+    12  0  01063+            LDA       LENGTH
          31  0  01016+            FSB       =1.0
```

```
00657+  20  0  01063+          STA      LENGTH
        75  0  00652+          SLJ      GETTOTAL
00660+  11  0  77776   MODE1CK  INA      -1
        22  1  00666+          AJP   N  MODE2CK
00661+  12  0  01160+          LDA      AMAX
        31  0  01163+          FSB      AMIN
00662+  22  3  00664+ +        AJP   M  *+2
        12  0  01160+          LDA      AMAX
00663+  30  0  01160+          FAD      AMAX
        75  0  00665+          SLJ      *+2
00664+  12  0  01163+ +        LDA      AMIN
        30  0  01163+          FAD      AMIN
00665+  20  0  01062+ +        STA      TOTAL
        75  0  00652+          SLJ      GETTOTAL
00666+  11  0  77776   MODE2CK  INA      -1
        22  1  00672+          AJP   N  MODE3CK
00667+  10  0  00000           ENA      0
        20  0  01163+          STA      AMIN
00670+  12  0  01160+          LDA      AMAX
        20  0  01062+          STA      TOTAL
00671+  75  0  00652+          SLJ      GETTOTAL
        50  0  00000
00672+  11  0  77776   MODE3CK  INA      -1
        22  1  00702+          AJP   N  MODE4CK
00673+  36  0  01040+ +        SSK      X/YFLAG
        75  0  00677+          SLJ      YMODE3
00674+  10  0  00000   XMODE3   ENA      0
        20  0  01163+          STA      AMIN
00675+  12  0  01160+          LDA      AMAX
        20  0  01062+          STA      TOTAL
00676+  75  0  00652+          SLJ      GETTOTAL
        50  0  00000
00677+  10  0  00000   YMODE3   ENA      0
        20  0  01160+          STA      AMAX
1 00700+  12  0  01163+          LDA      AMIN
        20  0  01062+          STA      TOTAL
00701+  75  0  00652+          SLJ      GETTOTAL
        50  0  00000
00702+  11  0  77776   MODE4CK  INA      -1
        22  1  00706+          AJP   N  MODE5CK
00703+  10  0  00000           ENA      0
        20  0  01160+          STA      AMAX
00704+  12  0  01163+          LDA      AMIN
        20  0  01062+          STA      TOTAL
00705+  75  0  00652+          SLJ      GETTOTAL
        50  0  00000
00706+  11  0  77776   MODE5CK  INA      -1
        22  1  00716+          AJP   N  MODERROR
00707+  36  0  01040+          SSK      X/YFLAG
```

```
                75  0  00713+              SLJ       YMODE5
00710+          10  0  00000    XMODE5     ENA       0
                20  0  01160+              STA       AMAX
00711+          12  0  01163+              LDA       AMIN
                20  0  01062+              STA       TOTAL
00712+          75  0  00652+              SLJ       GETTOTAL
                50  0  00000
00713+          10  0  00000    YMODE5     ENA       0
                20  0  01163+              STA       AMIN
00714+          12  0  01160+              LDA       AMAX
                20  0  01062+              STA       TOTAL
00715+          75  0  00652+              SLJ       GETTOTAL
                50  0  00000
00716+          10  0  00000    MODERROR   ENA       0
                75  0  00654+              SLJ       MODEOCK
00717+          75  0  77777    ENCODE     SLJ       **
                50  0  00000
00720+          10  0  01030+ +           ENA       =63
                04  0  01055+             ENQ       F.20
00721+          75  4  X00006             RTJ       INGOUT
                50  0  00000
00722+          00  0  01170+ +           ZRO       ITEMP
                00  0  01023+             ZRO       =10
00723+          75  4  X00004  +          RTJ       ERROR
                50  0  00000
00724+          75  4  X00007  +          RTJ       GOUT
                50  0  00000
00725+          00  0  01014+ +           ZRO       =0
                00  0  01072+             ZRO       X/YSCALE
00726+          75  4  X00004  +          RTJ       ERROR
                50  0  00000
00727+          75  4  X00010  +          RTJ       ENDGOUT
                50  0  00000
00730+          75  4  X00004  +          RTJ       ERROR
                50  0  00000
00731+          75  0  00717+ +           SLJ       ENCODE
                50  0  00000
00732+          75  0  77777    ALLZEROS   SLJ       **
                61  0  00734+             SAL       *+2
00733+          07  0  00060              LLS       48
                61  0  00735+             SAL       *+2
1       00734+  12  0  01015+             LDA       =1.0
                20  0  77777              STA       **
00735+          12  0  01015+             LDA       =1
                20  0  77777              STA       **
00736+          75  0  00732+             SLJ       ALLZEROS
                50  0  00000
00737+          75  0  77777    TPCHK      SLJ       **                  CHECK TAPE WRITE
                56  1  00765+             SIU    1  TPE
```

```
00740+  52  1  00737+          LIU   1  TPCHK
        51  1  77774           INI   1  -3
00741+  12  1  00000           LDA   1  0
        20  0  00753+          STA      TPPAR2
00742+  10  0  00060    TP1    ENA      48
        75  4 X00012           RTJ      IFUNIT          CAL IFUNIT TO CHECK WRITE
00743+  41  0  00770+ +         SCL      IFMSK
        22  3  00742+          AJP   3  TP1
00744+  22  0  00765+          AJP   0  TPE                   ZRO.  O.K., OUT
        05  0  00001           ALS   1
00745+  22  3  00756+          AJP   3  TPEOT                 BIT 47 NEG.=EOT
        72  0  00767+          RAO      TPCTR           PARITY ERROR
00746+  22  0  00765+          AJP   0  TPE                   TRY TO REWRITE 3 TIMES
        50  0  00000
00747+  10  0  01027+ +         ENA      =48                   IF NOT CORRECTED,
        75  4 X00013           RTJ      BKSP                  THEN IGNORE IT
00750+  75  4 X00015   +        RTJ      ERROR*
        50  0  00000
00751+  10  0  01015+  TPPAR1   ENA      =1                    REWRITE, TRY 3 TIMES
        04  0  01027+          ENQ      =48
00752+  75  4 X00005           RTJ      BUFOUT
        50  0  00000
00753+  00  0  77777   TPPAR2   ZRO      **
        00  0  77777           ZRO      **
00754+  75  4 X00015           RTJ      ERROR*
        50  0  00000
00755+  75  0  00742+ +         SLJ      TP1
        50  0  00000
00756+  05  0  00001   TPEOT    ALS   1  1                  EOT
        22  3  00761+          AJP   3  TPEOTP                LAST RECORD OK
00757+  10  0  01027+          ENA      =48
        75  4  00771+          RTJ      NEWTAPE               GET NEW TAPE
00760+  75  0  00765+ +         SLJ      TPE                   OUT
        50  0  00000
00761+  10  0  01027+  TPEOTP   ENA      =48           EOT WITH BAD PARITY
        75  4 X00013           RTJ      BKSP
00762+  75  4 X00015   +        RTJ      ERROR*
        50  0  00000
00763+  10  0  01027+ +         ENA      =48
        75  4  00771+          RTJ      NEWTAPE               GET NEW TAPE
00764+  75  0  00751+ +         SLJ      TPPAR1                REWRITE LAST RECORD
        50  0  00000
00765+  50  1  77777   TPE      ENI   1  **
        10  0  77774           ENA      -3            EXIT
00766+  20  0  00767+          STA      TPCTR                 RESET COUNTER TO TRY 3
        75  0  00737+          SLJ      TPCHK
00767+  77  7  77777   TPCTR    OCT      -3
        77  7  77774
1       00770+  07  7  77777   IFMSK    OCT      0777777777777777
```

```
                77   7   77777

00771+          75   0   77777     NEWTAPE  SLJ       **                        CALLED WHEN EOT FOUND
                56   1   01006+              SIU    1  NEWTPE                         WRITES DOUBLE END OF FILE
00772+          10   0   01027+              ENA       =48
                75   4  X00011              RTJ       WREOF                     WRITE TWO EOFS
00773+          75   4  X00015    +         RTJ       ERROR*
                50   0   00000
00774+          10   0   01027+   +         ENA       =48
                75   4  X00011              RTJ       WREOF
00775+          75   4  X00015    +         RTJ       ERROR*
                50   0   00000
00776+          04   0   00000    +         ENQ       0
                50   0   00000
00777+          75   4  X00014    +         RTJ       WRITE*
                01   0   00060              1         48
01000+          10   0   00000              10        0                         REWIND WITH LOCKOUT
                00   0   00001              ZRO       1
01001+          50   0   00000              NOP
                50   0   00000              NOP
01002+          04   0   00000    +         ENQ       0
                50   0   00000
01003+          75   4  X00014    +         RTJ       WRITE*                    TYPE MESAGE
                02   0   00066              2         54
01004+          03   0   01007+   +         3         MESAGE
                00   0   01014+              0         MESAGE+5
01005+          75   4  X00015    +         RTJ       ERROR*
                50   0   00000
01006+          50   1   77777     NEWTPE   ENI    1  **
                76   0   00771+              SLS       NEWTAPE                   WAIT FOR NEW TAPE
01007+          45   2   00301     MESAGE   FLX       5-R-EOT TAPE 48, MOUNT NEW TAPE AND START -R-
                04   0   13015
01010+          20   0   46233
                46   0   40703
01011+          34   0   60104
                06   2   03104
01012+          01   3   01520
                04   3   00622
01013+          04   2   40130
                12   0   10445
01014+          00   0   00000    =0       OCT       0
                00   0   00000
01015+          00   0   00000    =1       DEC       1
                00   0   00001
01016+          20   0   14000    =1.0     DEC       1.0
                00   0   00000
01017+          00   0   00000    =2       DEC       2
                00   0   00002
```

```
01020+  20  0  24000   =2.0       DEC   2.0
        00  0  00000
01021+  00  0  00000   =8         DEC   8
        00  0  00010
01022+  00  0  00000   =9         DEC   9
        00  0  00011
1       01023+  00  0  00000   =10        DEC   10
        00  0  00012
01024+  20  0  45000   =10.0      DEC   10.0
        00  0  00000
01025+  00  0  00000   =20        DEC   20
        00  0  00024
01026+  00  0  00000   =25        DEC   25
        00  0  00031
01027+  00  0  00000   =48        DEC   48
        00  0  00060
01030+  00  0  00000   =63        DEC   63
        00  0  00077
01031+  00  0  00000   =70        DEC   70
        00  0  00106
01032+  00  0  00000   =80        DEC   80
        00  0  00120
01033+  00  0  00000   =100       DEC   100
        00  0  00144
01034+  20  0  76200   =100.0     DEC   100.0
        00  0  00000
01035+  00  0  00000   =900       DEC   900
        00  0  01604
01036+  37  7  73777   NFLAG      OCT   3777377700000000
        00  0  00000
01037+  00  0  00000   INFLAG     OCT   37773777
        37  7  73777
01040+                 X/YFLAG    BSS   1
01041+                 O/UFLAG    BSS   1
01042+  01  0  20304   LABELCEL   BCD   112345678
        05  0  60710
01043+  00  0  00000   LABELCYC   OCT   0
        00  0  00000
01044+  00  7  70000   MK1        OCT   0077000000000000
        00  0  00000
01045+  00  0  00000   MK2        OCT   7777
        00  0  07777
01046+  77  7  77777   MASK       OCT   7777777700000000
        00  0  00000
01047+  00  0  00000   LOWMASK    OCT   77700000
        77  7  00000
01050+  77  7  77777   CLEARMK    OCT   7777777777700000
        77  7  00000
01051+  52  5  25252   CONTROL2   OCT   5252525252525252
```

```
                52  5  25252
01052+          52  5  25252  CONTR1    OCT       5252525252525252
                52  5  25252
01053+          00  0  00001  LOWMARG   DEC       1
                00  0  00001
01054+          02  0  00000  SCALSIZ   OCT       0200000000000000
                00  0  00000
01055+          34  6  50112  F.20      BCD       1(E10.2)
                73  0  27420
01056+          20  2  72061  SCALEX1   BCD       1 X AXIS
                27  7  12220
01057+          22  6  36143  SCALEX2   BCD       1SCALE =
                65  2  01320
1       01060+  20  3  02061  SCALEY1   BCD       1 Y AXIS
                27  7  12220
01061+          22  6  36143  SCALEY2   BCD       1SCALE =
                65  2  01320
01062+          00  0  00000  TOTAL     OCT       0
                00  0  00000
01063+          00  0  00000  LENGTH    OCT       0
                00  0  00000
01064+          00  0  00000  WIDTH     OCT       0
                00  0  00000
01065+          00  0  00000  HEIGHT    OCT       0
                00  0  00000
01066+          00  0  00000  EXPONENT  OCT       0
                00  0  00000
01067+          00  0  00000  MODEADDR  OCT       0
                00  0  00000
01070+          00  0  00000  QUANTA    OCT       0
                00  0  00000
01071+          00  0  00000  TEMPSCAL  OCT       0
                00  0  00000
01072+          00  0  00000  X/YSCALE  OCT       0
                00  0  00000
01073+          00  0  00000  TEMP      OCT       0
                00  0  00000
01074+          00  0  00000  IOFFS     OCT       0
                00  0  00000
01075+          00  0  00000  IXOFFS    OCT       0
                00  0  00000
01076+          00  0  00000  IYOFFS    OCT       0
                00  0  00000
01077+          00  0  00000  IWIDE     OCT       0
                00  0  00000
01100+          00  0  00000  LABELX    OCT       0
                00  0  00000
01101+          00  0  00000  LABELY    OCT       0
                00  0  00000
```

```
01102+  00  0  00000  XSCALE    OCT        0
        00  0  00000
01103+  00  0  00000  YSCALE    OCT        0
        00  0  00000
01104+  00  0  00000  XSCLAB1   OCT        0
        00  0  00000
01105+  00  0  00000  XSCLAB2   OCT        0
        00  0  00000
01106+  00  0  00000  YSCLAB1   OCT        0
        00  0  00000
01107+  00  0  00000  YSCLAB2   OCT        0
        00  0  00000
01110+  00  0  00000  MODCURV   OCT        0
        00  0  00000
01111+  00  0  00000  NUMPTS    OCT        0
        00  0  00000
01112+  00  0  00000  X1ADDR    OCT        0
        00  0  00000
01113+  00  0  00000  Y1ADDR    OCT        0
        00  0  00000
1       01114+  00  0  00000  SFXADDR   OCT        0
        00  0  00000
01115+  00  0  00000  SFYADDR   OCT        0
        00  0  00000
01116+  00  0  00000  IHIGH     OCT        0
        00  0  00000
01117+  00  0  00000  TITLESIZ  OCT        0
        00  0  00000
01120+  00  0  00000  T1/2ADDR  OCT        0
        00  0  00000
01121+  00  0  00000  INTERPS   OCT        0
        00  0  00000
01122+  00  0  00000  INCRJVL   OCT        0
        00  0  00000
01123+  00  0  00000  IVH       OCT        0
        00  0  00000
01124+  00  0  00000  INCRVH    OCT        0
        00  0  00000
01125+  00  0  00000  NH        OCT        0
        00  0  00000
01126+  00  0  00000  IVL       OCT        0
        00  0  00000
01127+  00  0  00000  JVL       OCT        0
        00  0  00000
01130+  00  0  00000  IVV       OCT        0
        00  0  00000
01131+  00  0  00000  INCRVV    OCT        0
        00  0  00000
01132+  00  0  00000  NV        OCT        0
```

```
                00   0  00000
01133+          00   0  00000  LSVOV     OCT        0
                00   0  00000
01134+          00   0  00000  IT        OCT        0
                00   0  00000
01135+          00   0  00000  JT        OCT        0
                00   0  00000
01136+          00   0  00000  IH        OCT        0
                00   0  00000
01137+          00   0  00000  JV        OCT        0
                00   0  00000
01140+          00   0  00000  JH        OCT        0
                00   0  00000
01141+          00   0  00000  JHL       OCT        0
                00   0  00000
01142+          00   0  00000  LH        OCT        0
                00   0  00000
01143+          00   0  00000  IHL       OCT        0
                00   0  00000
01144+          00   0  00000  INCRIHL   OCT        0
                00   0  00000
01145+          00   0  00000  IV        OCT        0
                00   0  00000
01146+          00   0  00000  LV        OCT        0
                00   0  00000
01147+          00   0  00000  ITOR      OCT        0
                00   0  00000
1       01150+  00   0  00000  LSHOH     OCT        0
                00   0  00000
01151+          00   0  00000  IPT       OCT        0
                00   0  00000
01152+          00   0  00000  JPT       OCT        0
                00   0  00000
01153+          00   0  00000  BUFAD     OCT        0
                00   0  00000
01154+          00   0  00000  KTITLE    OCT        0
                00   0  00000
01155+          00   0  00000  LABEL     OCT        0
                00   0  00000
01156+          00   0  00000  XMAX      OCT        0
                00   0  00000
01157+          00   0  00000  AXMAX     OCT        0
                00   0  00000
01160+          00   0  00000  AMAX      OCT        0
                00   0  00000
01161+          00   0  00000  XMIN      OCT        0
                00   0  00000
01162+          00   0  00000  AXMIN     OCT        0
                00   0  00000
```

```
01163+  00  0  00000   AMIN     OCT      0
        00  0  00000
01164+  00  0  00000   YMAX     OCT      0
        00  0  00000
01165+  00  0  00000   AYMAX    OCT      0
        00  0  00000
01166+  00  0  00000   YMIN     OCT      0
        00  0  00000
01167+  00  0  00000   AYMIN    OCT      0
        00  0  00000
01170+                ITEMP    BSS      15
01207+                STORE    BSS      460
                               END

   NO DOUBLY DEFINED
   NO UNDEFINED SYMBOLS
   NO ERRORS
      NULLS                          CALCVALS          XMODE3          XMODE5          =80
                  00000                END       PLOTEST
 EXECUTE,100,56,0.
1 PROGRAM NAMES
0 00000 RESIDENT                      02002 ERRDUMP*                  72252 PLOTEST
  70677 GOUT                          70625 ERROR                     70602 FORTNTRY
  70530 FORTFLGS                      66405 GRAFPLOT                  66374 FLOATF
  66323 XTOI                          66257 XINTF                     66114 BUFINOUT
  66073 WRITEEOF                      66022 BACKSPAS
0 LABELED COMMON          NONE
0 NUMBERED COMMON         NONE
0 PROGRAM ENTRY POINTS
0 01205 READ*                         01212 WRITE*                    00655 SELECT*
  01140 REMOVE*                       00743 DETECT*                   01760 GETCH*
  01765 CHKSTD*                       03044 LOADER*                   03045 RELOCOM*
  02735 LIBREW*                       02654 CLBNBCD*                  02613 EXIT*
  01774 ERROR*                        01174 MODIRET*                  00401 BCDBN*
  00335 OPCOM*                        02227 RECRET*                   02246 RECLIM*
  00524 FLAGTST*                      00217 MEMREC*                   01737 CBF*
  01740 CBF1*                         01741 CBF2*                     01746 CBF7*
  00714 IOSLECT*                      01470 MRW600*                   05322 LOK1*
  04730 LMSRCH*                       01121 ILMF*                     01747 FCH*
  01034 EXSEN*                        00167 SETCLK*                   00076 DATE*
  00061 SIEOF                         02727 LIBPOSIT                  72252 PLOTEST
  70677 INGOUT                        71144 GOUT                      71123 ENDGOUT
  70625 ERROR                         70650 ERROR2                    66427 GRAFPLOT
  70602 FORTNTRY                      70605 FORTEXIT                  70621 FORTGIN
  70622 FORTGOUT                      70530 FSETEOFS                  70533 FSETIOCK
  70536 FSETTAPE                      70555 FSENEOFS                  70561 FSENIOCK
  70565 FSENTAPE                      66405 AUTOGRAF                  66404 FLOATF
  66323 XTOI                          66257 XINTF                     66114 BUFIN
  66120 BUFOUT                        66230 IFUNIT                    66073 WREOF
  66022 BKSP
```

```
      PROGRAM SORT (IA, IB, IC, ID, IE, IF)                                    00010
                                                                               00020
*     LAST CORRECTED 30 MARCH 1963,  R.L. HOGG                                 00030
*     NOTE -- THIS PROGRAM IS FOR TYPEWRITER CALL                              00040
                                                                               00050
                                                                               00060
      LOC(BCDTYPE=25, LOCK=33)                                                 00070
      HOL(FINISH0= SORTED ,FINISH1=OUTPUT I,FINISH2=S ON TAP,                  00080
     *    FINISH3=E NR.    )                                                   00090
      CON(MKSOUT=777777777770000B)                                             00100
      CON(ERROR=2020714547242320B,ERROR1=615167206551514 6B,                   00110
     *    ERROR2=5100000000000000B)                                           00120
      ENA(IA)           STA(ICOL)          .                                   00130
      ENA(IB)           STA(IICOL)         .                                   00140
      ENA(IC)           STA(INPUT)         .                                   00150
      ENA(ID)           STA(ITAPE)         .                                   00160
      ENA(IE)           STA(IITAPE)        .                                   00170
      ENA(IF)           STA(LISTOUT)       .                                   00180
      EXF(62540B)       SLJ1(12)           . LOCKOUT SAT. / JK1 , GO DUMP      00190
      IF(ICOL-IICOL) 8,8,13                                                    00200
 8    LAST=IICOL                                                               00210
 9    INCR=6                                                                   00220
 10   INIT=LAST-INCR                                                           00230
      IF(INIT-ICOL) 11,12,12                                                   00240
 11   INCR=INCR-1                                                              00250
      GO TO 10                                                                 00260
 12   CALL SORT7 (INIT,LAST,INPUT,ITAPE,IITAPE,LISTOUT)                        00270
      IF(INIT-ICOL) 13,15,14                                                   00280
 13   ENA(ERROR)        SLJ4(BCDTYPE)    .                                     00290
      GO TO 16                                                                 00300
 14   INPUT=LISTOUT                                                            00310
      LAST=INIT-1                                                              00320
      GO TO 9                                                                  00330
 15   LDA(LOCK)         INA(1)           . GET ADDRESS FOR LOCK ARG            00340
      SAL(L+1)          ENI(0)           .                                     00350
      LDA(LISTOUT)      STA(N)           . SET LOCK ARG                        00360
      SLJ4(LOCK)        ZRO(0)           .                                     00370
      LDA(LISTOUT)      ALS(6)           .                                     00380
      LDQ(MKSOUT)       SSU(FINISH3)     .                                     00390
      STA(FINISH3)      ENA(FINISH0)     .                                     00400
      SLJ4(BCDTYPE)     ZRO(0)           .                                     00410
 16   END                                                                      00420
                                                                               00430
                                                                               00440
      SUBROUTINE SORT7 (ICOL,IICOL,INPUT,NFLAG3,NFLAG4,LISTOUT)                00450
C     SORT7  CAN ONLY SORT ON SEVEN COLUMNS AT A TIME                          00460
C     ICOL = LOWEST NR. COLUMN OF FIELD                                        00470
```

```
C      IICOL = HIGHEST NR. COLUMN OF FIELD, MAX FIELD WIDTH = 7                  00480
C      NFLAG1 = LAST BLOCK KEY WORD COUNT                                        00490
C      NFLAG2 = RUNNING TOTAL OF NR. OF BLOCKS ON MERGED TAPE                    00500
C      NFLAG3 = TAPE NR. WHICH HOLDS MERGED FILE                                 00510
C      NFLAG4 = TAPE NR. ON WHICH TO ORDER NEXT MERG                             00520
       SLJ1(400)                                                                 00530
       NFLAG1 = -0                                                               00540
       NFLAG2 = 0                                                                00550
       REWIND INPUT                                                              00560
       REWIND NFLAG3                                                             00570
       REWIND NFLAG4                                                             00580
       REWIND LISTOUT                                                            00590
       NRCOLS=IICOL-ICOL+1                                                       00600
       IF(NRCOLS-7) 100,100,99                                                   00610
    99 STOP                                                                      00620
   100 CALL ALSINP(ICOL,IICOL,INPUT)                                             00630
       ALS(33)          ARS(33)          . EXTEND SIGN BIT                      00640
       STA(NFLAG1)      AJP3(200)        .                                      00650
       AJP(260)                                                                  00660
   200 CALL RLHMERG (NFLAG1)                                                     00670
       IF(NFLAG2) 250,250,300                                                    00680
   250 CALL WRITE (NFLAG3,37000B,57000B,1)                                       00690
       NFLAG2 = 1                                                                00700
       REWIND NFLAG3                                                             00710
       LDA(NFLAG1)    AJP3(100)                                                  00720
       SLJ(400)                                                                  00730
   260 NFLAG2=NFLAG2+1                                                           00740
       SLJ(400)                                                                  00750
   300 CALL HEXSORT(NFLAG1,NFLAG2,NFLAG3,NFLAG4)                                 00760
       NFLAG2 = NFLAG2 + 1                                                       00770
       ITEMP = NFLAG3                                                            00780
       NFLAG3 = NFLAG4                                                           00790
       NFLAG4=ITEMP                                                              00800
       REWIND NFLAG3                                                             00810
       REWIND NFLAG4                                                             00820
       LDA(NFLAG1)    AJP3(100)                                                  00830
   400 REWIND INPUT                                                              00840
       REWIND NFLAG4                                                             00850
       CALL KREMKEY(NFLAG1,NFLAG2,NFLAG3,LISTOUT)                                00860
       END                                                                       00870
                                                                                 00880
                                                                                 00890
                                                                                 00900
       MACHINE RDMT(1ARG,2ARG,3ARG,4ARG,5ARG,6ARG)                               00910
                                                                                 00920
                                                                                 00930
* NOTE  G = ADDRESS OF ENTRY TO TYPEWRITER SENSE ROUTINE                         00940
*       P = ADDRESS OF PROGRAM CONTROL ROUTINE JUMP  ENTRY                       00950
* THIS ROUTINE IS FOR PROGRAM CALL                                              00960
```

```
      LOC(G=31,  P=10)                                                          00970
                                                                                00980
                                                                                00990
                                                                                01000
* TYPEWRITER MESSAGE CODE TABLE                                                 01010
                                                                                01020
       CON(T6 =4504041220302204B,  T7 =1120061301050420B,                       01030
      *    T8 =1212031200000000B,  T9 =4504041220302204B,                       01040
      *    T10=1530121401250420B,  T11=1212031200000000B)                       01050
       CON(T12=4504041220302204B,  T13=2006220403260426B,                       01060
      *    T14=1411200000000000B,  T15=4504043112140120B,                       01070
      *    T40=4504040130152004B,  T41=3406140104060301B,                       01080
      *    T42=0414062214163001B,  T43=2022420000000000B)                       01090
                                                                                01100
                                                                                01110
* TAPE UNIT ASSIGNMENT TABLE                                                    01120
*      ASSIGNMENTS FOR CHANNEL 3/4 LIBRARY TAPE                                 01130
                                                                                01140
       CON(K0 =0       ,  K1 =332010B,  K2 =332020B,                            01150
      *    K3 =332030B,  K4 =332040B,  K5 =552010B,                            01160
      *    K6 =552020B,  K7 =552030B,  K8 =552040B)                            01170
                                                                                01180
                                                                                01190
* ASSORTED CONSTANTS                                                            01200
                                                                                01210
       CON(R4 =7373737373737373B,  R16=0070000000000000B)                       01220
                                                                                01230
                                                                                01240
* TAPE READING ROUTINE                                                         01250
                                                                                01260
1X     SLJ(N)          SLJ(L+7)        . EXIT/ENTRY                           01270
1ARG   ZRO(0)          ZRO(0)          . TAPE UNIT NUMBER ARG.                01280
2ARG   ZRO(0)          ZRO(0)          . INITIAL ADDRESS                      01290
3ARG   ZRO(0)          ZRO(0)          . TERMINAL ADDRESS                     01300
4ARG   ZRO(0)          ZRO(0)          . MODE, 1 = BINARY, 2 = BCD            01310
5ARG   ZRO(0)          ZRO(0)          .                                      01320
6ARG   ZRO(0)          ZRO(0)          . ENDFILE EXIT ADDRESS LOCATOR         01330
       LDA(5ARG)       SAU(22X)        .                                      01340
       LDA7(6ARG)      SAL(23X)        .                                      01350
       SIU1(23X)       ENI(0)          .                                      01360
       LDA7(1ARG)      INA(-11B)       .                                      01370
       SIU1(20X)       AJP3(L+3)       . CHECK FOR TU ASSIGNMENT              01380
       ENA(T40)        SLJ4( 3)        . GO PRINT ERROR                       01390
       SLJ( P)         ZRO(0)          .                                      01400
       INA(11B)        STA(K0)         .                                      01410
       LIL1(K0)        LDA1(K0)        .                                      01420
       ADD7(4ARG)      SAU(4X)         . SET TU CONTROL CODES                 01430
       SCL(777B)       SAL(4X)         . INITIALIZE                           01440
       SAL(5X)         SAL(11X)        .                                      01450
```

```
      INA(3)          SAU(8X)       .                                  01460
      INA(2)          SAU(9X)       .                                  01470
      INA(1)          SAU(11X)      .                                  01480
      INA(1)          SAU(7X)       .                                  01490
      LRS(18)         ENA(0)        .                                  01500
      LLS(3)          SAL(2X)       .                                  01510
      SAL(3X)         SAU(12X)      .                                  01520
      LLS(39)         LDQ(R16)      .                                  01530
      STA(K0)         LDA(5X)       .                                  01540
      SSU(K0)         STA(5X)       .                                  01550
      LDA7(2ARG)      SAU(5X)       .                                  01560
      LDA(6X)         SSU(K0)       .                                  01570
      STA(6X)         ENI(0)        .                                  01580
2X    LDA7(3ARG)      SAL(N)        .                                  01590
      SAU(6X)         SIU2(19X)     .                                  01600
3X    ENI2(4)         LIL1(N)       .                                  01610
4X    EXF(N)          EXF7(N)       . SELECT UNIT                      01620
5X    EXF(N)          EXF7(N)       . ACTIVATE CHANNEL                 01630
6X    EXF(N)          LDA7(2ARG)    .                                  01640
      SUB(R4)         LLS(48)       .                                  01650
      INI1(-1)        LDA1(Z)       .                                  01660
      INI1(1)         SUB(R4)       .                                  01670
7X    EXF7(N)         SLJ(21X)      . SENSE ENDFILE                    01680
8X    EXF7(N)         SLJ(L+4)      . SENSE PARITY ERROR               01690
9X    EXF7(N)         SLJ(10X)      . SENSE LENGTH ERROR               01700
      QJP1(19X)       AJP(3X)       . SENSE FOR BAD RECORD             01710
      SLJ(19X)        ZRO(0)        . EXIT                             01720
      IJP2(L+2)       SLJ(14X)      . SENSE FIVE PARITY ERRORS         01730
10X   IJP2(L+1)       SLJ(13X)      . SENSE FIVE LENGTH ERRORS         01740
      AJP(3X)         QJP(3X)       . SENSE FOR BAD RECORD             01750
11X   EXF(N)          EXF7(N)       . BACKSPACE                        01760
12X   LIL1(N)         SLJ(4X)       . RETURN TO REREAD                 01770
13X   ENA(T6)         SLJ(17X)      . SET TO INDICATE TROUBLE          01780
14X   ENA(T9)         SLJ(17X)      .                                  01790
15X   ENA(T12)        SLJ(18X)      .                                  01800
17X   SLJ4( G)        ZRO(0)        . TROUBLE EXIT                     01810
      AJP(19X)        AJP3(3X)      . SENSE ACTION TO TAKE             01820
      SLJ(11X)        ZRO(0)        .                                  01830
18X   SLJ4( G)        ZRO(0)        . TROUBLE EXIT                     01840
      AJP(3X)         AJP3(3X)      . SENSE ACTION TO TAKE             01850
      SLJ(11X)        ZRO(0)        .                                  01860
19X   ENI2(N)         ENI(0 )       . RE-STORE INDEXES                 01870
20X   ENI1(N)         SLJ(1X)       .                                  01880
21X   LIU2(L-2)       INI2(-1)      .                                  01890
22X   SIL2(N)         ENI2(512)     .                                  01900
23X   ENI1(N)         SLJ(N)        . ENDFILE DETECTED/EXIT            01910
      END                                                              01920
                                                                       01930
                                                                       01940
```

```
SUBROUTINE ALSINP(ICOL,IICOL,INPUT)                                    01950
LOC(INT=00005, TYPE=31)                                                01960
CON(MSK1=4000000000000008)                                             01970
CON(ERR0 =4504044714111120B,      ERR1 =1330110423162204B)             01980
CON(ERR2 =2425072303110414B,      ERR3 =0616033406012012B)             01990
CON(ERR4 =2022404504042434B,      ERR5 =2324011401342220B)             02000
CON(ERR6 =2204300462420404B,      ERR7 =1512202424041612B)             02010
CON(ERR10=0401030416030501B,      ERR11=1406342042570000B)             02020
CON(BCD0 =5353535353535353B,      BCD1 =7474747474747474B)             02030
CON(BCD2 =3434343434343434B,      BCD3 =1414141414141414B)             02040
CON(BCD4 =3333333333333333B,      BCD5 =2121212121212121B)             02050
CON(BCD6 =1313131313131313B,      BCD7 =6060606060606060B)             02060
CON(BCD10=4040404040404040B,      BCD11=5454545454545454B)             02070
CON(BCD12=7373737373737373B,      BCD13=3131313131313131B)             02080
CON(BCD14=3030303030303030B,      BCD15=2727272727272727B)             02090
CON(BCD16=2626262626262626B,      BCD17=2525252525252525B)             02100
CON(BCD20=2424242424242424B,      BCD21=2323232323232323B)             02110
CON(BCD22=2222222222222222B,      BCD23=5151515151515151B)             02120
CON(BCD24=5050505050505050B,      BCD25=4747474747474747B)             02130
CON(BCD26=4646464646464646B,      BCD27=4545454545454545B)             02140
CON(BCD30=4444444444444444B,      BCD31=4343434343434343B)             02150
CON(BCD32=4242424242424242B,      BCD33=4141414141414141B)             02160
CON(BCD34=7171717171717171B,      BCD35=7070707070707070B)             02170
CON(BCD36=6767676767676767B,      BCD37=6666666666666666B)             02180
CON(BCD40=6565656565656565B,      BCD41=6464646464646464B)             02190
CON(BCD42=6363636363636363B,      BCD43=6262626262626262B)             02200
CON(BCD44=6161616161616161B,      BCD45=1111111111111111B)             02210
CON(BCD46=1010101010101010B,      BCD47=0707070707070707B)             02220
CON(BCD50=0606060606060606B,      BCD51=0505050505050505B)             02230
CON(BCD52=0404040404040404B,      BCD53=0303030303030303B)             02240
CON(BCD54=0202020202020202B,      BCD55=0101010101010101B)             02250
CON(BCD56=1212121212121212B,      BCD57=2020202020202020B)             02260
CON(ORD0 =5757575757575757B,      ORD1 =5656565656565656B)             02270
CON(ORD2 =5555555555555555B,      ORD3 =5454545454545454B)             02280
CON(ORD4 =5353535353535353B,      ORD5 =5252525252525252B)             02290
CON(ORD6 =5151515151515151B,      ORD7 =5050505050505050B)             02300
CON(ORD10=4747474747474747B,      ORD11=4646464646464646B)             02310
CON(ORD12=4545454545454545B,      ORD13=4444444444444444B)             02320
CON(ORD14=4343434343434343B,      ORD15=4242424242424242B)             02330
CON(ORD16=4141414141414141B,      ORD17=4040404040404040B)             02340
CON(ORD20=3737373737373737B,      ORD21=3636363636363636B)             02350
CON(ORD22=3535353535353535B,      ORD23=3434343434343434B)             02360
CON(ORD24=3333333333333333B,      ORD25=3232323232323232B)             02370
CON(ORD26=3131313131313131B,      ORD27=3030303030303030B)             02380
CON(ORD30=2727272727272727B,      ORD31=2626262626262626B)             02390
CON(ORD32=2525252525252525B,      ORD33=2424242424242424B)             02400
CON(ORD34=2323232323232323B,      ORD35=2222222222222222B)             02410
CON(ORD36=2121212121212121B,      ORD37=2020202020202020B)             02420
                                                                       02430
```

```
      CON(ORD40=1717171717171717B,     ORD41=1616161616161616B)          02440
      CON(ORD42=1515151515151515B,     ORD43=1414141414141414B)          02450
      CON(ORD44=1313131313131313B,     ORD45=1212121212121212B)          02460
      CON(ORD46=1111111111111111B,     ORD47=1010101010101010B)          02470
      CON(ORD50=0707070707070707B,     ORD51=0606060606060606B)          02480
      CON(ORD52=0505050505050505B,     ORD53=0404040404040404B)          02490
      CON(ORD54=0303030303030303B,     ORD55=0202020202020202B)          02500
      CON(ORD56=0101010101010101B,     ORD57=0000000000000000B)          02510
      ENA(1EFX)          STA(ENDFILE)      .                             02520
      ENA(-0)            STA(SWITCH)       .                             02530
      STA(SLK1)                            .                             02540
      ENA(17000B)        STA(KEY)          .                             02550
      INA(1)             STA(INIT)         .                             02560
      SAU(2WHL)          SAU(2TRU)         .                             02570
      INA(17B)           STA(ITERM)        .                             02580
1CKK  SIU2(2STI)     ENI2(1)          .SAVE B2 AND SET TO ONE           02590
      LDA(IICOL)     SUB(ICOL)        .WINDOW SHIFTS                    02600
      SAU(2CNV)      SIL6(2STI)       .                                 02610
      INA(1)         MUI(6)           .SHIFT COUNT =6XNO. CHAR          02620
      SAL(2CKK)      ENQ(-0)          .ALL ONES TO Q                    02630
2CKK  ENA(0)         LLS(N)           .CLEAR A AND SHIFT AQ LEFT TO     02640
      STQ(MSK2)      ENQ(0)           .FORM MASK FOR SELECTIVE CLEARING 02650
      LDA(ICOL)      DVI(8)           .                                 02660
      QJP1(L+1)        STQ(SWITCH)    .                                 02670
      STA(TMP1)      LDA(IICOL)       .CHECK BOTH ENDS OF EKEY          02680
      ENQ(0)                          .SET UP AQ FOR ANOTHER DVI        02690
      DVI(8)         STQ(TMP2)        .                                 02700
      SUB(TMP1)      AJP1(1SPL)       .JUMP IF KEY IS IN TWO WORDS      02710
      SSK(SWITCH)      SLJ(1TRU)      .                                 02720
1WHL  ENA(8)         SUB(TMP2)        .EIGHT MINUS R2                   02730
      MUI(6)         SAL(3WHL)        .PLACE SHIFT COUNT                02740
2WHL  ENA(N)         ADD(TMP1)        .GET ADDRESS OF WORD WITH KEY     02750
      SAU(3WHL)      ENI(0)           .RETURN JUMP TO READ NEXT RECORD  02760
      CALL RDMT(INPUT,INIT,ITERM,2,SLK1,ENDFILE)                        02770
3WHL  LDA(N)         ARS(N)           .RIGHT JUSTIFY KEY                02780
      SLJ4(1CNV)                      .RETURN JUMP TO CONVERT KEY       02790
      SLJ(2WHL)                       .PREPARE FOR NEXT RECORD          02800
1SPL  LDA(TMP2)      AJP1(1TRJ)       .KEY REALLY SPLIT                 02810
      ENA(8)         STA(TMP2)        .CHANGE R2 TO EIGHT               02820
      SLJ(1WHL)                       .KEY NOT SPLIT AFTER ALL          02830
1TRU  ENA(6)         MUI(TMP2)        .SIX TIMES R2                     02840
      SAU(4TRU)                       .PLACE SHIFT COUNT FOR LLS        02850
2TRU  ENA(N)         ADD(TMP1)        .COMPUTE CELL WITH START KEY      02860
      SAU(3TRU)      INA(1)           .CELL WITH ENDKEY                 02870
      SAL(3TRU)      ENI(0)           .RETJUMP TO READ NEXT RECORD      02880
      CALL RDMT(INPUT,INIT,ITERM,2,SLK1,ENDFILE)                        02890
3TRU  LDA(N)         LDQ(N)           .KEY TO AQ , RIGHT JUSTIFY IN A   02900
4TRU  LLS(N)         SLJ4(1CNV)       .RETJUMP TO CONVERT KEY           02910
      SLJ(2TRU)                       .PREPARE FOR NEXT RECORD          02920
```

```
 1CNV SLJ(N)            SIL3(3STI)       .                                  02930
 2CNV ENI3(N)           SCL(MSK2)        .                                  02940
      STA(TEMPKEY)      ENQ(773)         .                                  02950
1LOOP  LDL(TEMPKEY)     ENI6(60B)        .                                  02960
      MEQ6(BCDO)        SLJ(1ERR)        .                                  02970
      LDA(TEMPKEY)      SSU6(ORDO)       .                                  02980
      STA(TEMPKEY)      QLS(6)           .                                  02990
 3STI IJP3(1LOOP)       ENI3(N)          .                                  03000
      LDA(TEMPKEY)      ENI(O)           .                                  03010
      SCM(MSK1)         STA7(KEY)        . COMPLIMENT 48TH BIT              03020
      ENA(20B)          RAD(KEY)         .                                  03030
      INA(1)            SAL(INIT)        .                                  03040
      SAU(2WHL)         SAU(2TRU)        .                                  03050
      INA(17B)          SAL(ITERM)       .                                  03060
1EFX  ISK2(512)         SLJ(1CNV)        .                                  03070
 2STI ENI2(N)           ENI6(N)          .                                  03080
      LDA(SLK1)         ENI(O)           .                                  03090
      RETURN                                                                03100
 1ERR SIU1(L+4)         STQ(TEMPQ)       .                                  03110
      LDA(TEMPKEY)      SSU(BCDO)        .                                  03120
      STA(TEMPKEY)      ENA(ERRO)        .                                  03130
      SLJ4(TYPE)        ZRO(O)           .                                  03140
      ENI1(N)           LDQ(TEMPQ)       .                                  03150
      QLS(6)            SLJ(3STI)        .                                  03160
      END                                                                   03170
                                                                            03180
                                                                            03190
                                                                            03200
      MACHINE RLHMERG (1A)                                                  03210
      RSV(RA=3B, VA=11B)                                                    03220
      LOC(ABIN=17000, BBIN=37000)                                           03230
      CON(KA=20B, KA1=17000B, KA2=20000B,                                   03240
     *    KA3=37000B, KA4=3777777777777777B)                               03250
1AA   SLJ(N)             SLJ(L+2)         . EXIT/ENTRY                       03260
      ZRO(0)             ZRO(0)           . NFLAG1                           03270
      LDA7(L-1)          STA(VA)                                             03280
1BA   SIU1(RA)           ENI1(0)          . SAVE                             03290
      SIL2(RA)           ENI2(0)          .    INDEXES                       03300
      SIU3(RA+1)         ENI3(0)          .       AND                        03310
      SIL4(RA+1)         ENI4(0)          .          SET                     03320
      SIU5(RA+2)         ENI5(0)          .             TO                   03330
      SIL6(RA+2)         ENI6(0)          .                ZERO              03340
      LDA(KA1)           STA(VA+1)        . ALPHA = 17,000B                  03350
      LDA(KA3)           STA(VA+2)        . BETA = 37,000B                   03360
      LDA(KA)            STA(VA+3)        . ORDSIZE = 20B                    03370
      LDA(KA2)           STA(VA+4)        . ALTERNATING SIGNED 20,000 = (+)  03380
      LDA(VA)            AJP3(1BC)        . TEST NFLAG1, IF NEG. GO TO CYCLE 03390
*     LASTBLC                                                               03400
1BB   MUI(KA)            STA(VA+7)        . CALC NR. OF WORDS IN LASTBLC     03410
```

```
        LIL6(VA+7)      ENI(0)          . SET B6=TOTAL NR.                      03420
        LDA(KA4)        STA6(A3IN)      .    MODIFY                             03430
        INI6(20B)       SIL6(VA+10B)    .       EXCESSIVE                       03440
        LDA(VA+10B)     SUB(KA2)        .          KEYS IN                      03450
        AJP3(1BB+2)     ENI(0)          .             BLOCK                     03460
*       CYCLE                                                                   03470
1BC     LDA(VA+2)       SAL(2BJ)        . SET FIRST ADDRESS THIS CYCLE          03480
        LDA(VA+1)       STA(VA+5)       . SET FIRSTL = ALPHA                    03490
        LDA(VA+3)       INA(77776B)     . GENERATE ORDSIZE - 1                  03500
        SAU(4BF)        SAU(4BN)        . SET(ORDSIZE) IN STOR,RRUNOUT          03510
        SAU(4BG)        SAU(4BM)        . SET(ORDSIZE) IN STORL AND LRUNOUT     03520
*       PRE MERG                                                                03530
1BD     LDA(VA+5)       SAU(1BE)        . SET FIRSTL ADDRESS IN MERG            03540
        ADD(VA+3)       STA(VA+6)       . GENERATE FIRSTR ADDRESS               03550
        SAL(1BE)        LDQ2(VA+6)      . SET FIRSTR ADDRESS IN MERG            03560
*       MERG                                                                    03570
1BE     LDA1(N)         SUB2(N)         . SUBTRACT CURRENT KEYR FROM KEYL       03580
        AJP(1BF)        ENI(0)          . IF DIFF IS ZERO, GO STORL             03590
        AJP2(1BG)       SLJ(1BF)        . IF DIFF IS (+) GO TO STORR,           03600
*                                       . IF(-) GO TO STORL                     03610
*       STORL                                                                   03620
1BF     LDA(VA+5)       INA1(0)         . GENERATE ADDRESS OF KEYL              03630
        SAU(2BJ)        SLJ4(13J)       . SET ADDR OF KEYL IN (STORE),          03640
*                                         RETURN JUMP TO (STORE)                03650
        INI1(17B)       ENI(0)          . INCREASE INDEX 1 BY 17B               03660
4BF     ISK1(N)         SLJ(1BE)        . IF B1=ORDSIZE-1, ZERO B1,EXIT         03670
*                                         IF NOT, B1 = B1 + 1 GOTO MERG         03680
        SLJ(1BM)        ZRO(0)          . COMPLETE STORAGE OF RIGHT LIST        03690
*       STORR                                                                   03700
1BG     LDA(VA+6)       INA2(0)         . GENERATE ADDRESS OF KEYR              03710
        SAU(2BJ)        SLJ4(13J)       . SET ADDR KEYR IN (STORE)RJ(STORE)     03720
        INI2(17B)       ENI(0)          . INCREASE B2, 17B                      03730
4BG     ISK2(N)         SLJ(1BE)        . IF B2=ORDSIZE-1,ZERO B2               03740
*                                         IF NOT, B2=B2+1, GO TO MERG           03750
        SLJ(1BN)        ZRO(0)          . COMPLETE STORAGE LEFT LIST            03760
*       NEW PAIR                                                                03770
1BH     LDA(VA+3)       ADD(VA+3)       . GENERATE PAIR SIZE                    03780
        ADD(VA+5)       STA(VA+5)       . GEN/SET ADDR OF FIRSTL                03790
        ENI1(0)         ENI2(0)         . ZERO B1 AND B2                        03800
        SLJ(1BD)        ENI(0)          . GO TO PRE-MERG                        03810
*       RE-CYCL                                                                 03820
1BI     LDA(VA+1)       ADD(VA+4)       . ALPHA= ALPHA (+) OR (-) 20,000B       03830
        STA(VA+1)       LAC(VA+4)       . COMPLIMENT (+) OR (-) 20,000B         03840
        STA(VA+4)       RAD(VA+2)       . BETA = BETA (+) OR (-) 20,000B        03850
        LDA(VA+3)       RAD(VA+3)       . DOUBLE ORDSIZE                        03860
        ISK5(10B)       SLJ(1BC)        . EXIT IF MERG COMPLETE                 03870
*                                         IF NOT GO TO CYCLE                    03880
        SLJ(1BK)        ZRO(0)          . GO TO (EXIT)                          03890
*       STORE                                                                   03900
```

```
1BJ   SLJ(N)           ENI3(0)        . ENTRY/EXIT,  ZERO B3                    03910
2BJ   LDA3(N)          STA4(N)        . STORE KEY AND                           03920
      INI4(1)          ENI(0)         .    TRAILER WORDS                        03930
      ISK3(17B)        SLJ(1BJ+1)     .       AT END OF MERGED LIST             03940
      INI4(77776B)     ENI(0)         . INCREASE B1 BY (-1)                     03950
      ISK4(17777B)     SLJ(1BJ)       . IF BLC PROCESS COMPLETE GO TO           03960
*                                        (RE-CYCLE)/ IF NOT COMPLETE IT         03970
      ENI1(0)          ENI2(0)        . ZERO  B1, B2                            03980
      SLJ(1BI)         ZRO(0)         . JUMP TO RE-CYCL                         03990
*     EXIT                                                                      04000
1BK   LDA(VA)          AJP2(13L)      . NFLAG1=-0,IF NOT GOTO(FINAL EXIT)       04010
      LIU1(RA)         LIL2(RA)       . RESET                                   04020
      LIU3(RA+1)       LIL4(RA+1)     .    ALL                                  04030
      LIU5(RA+2)       LIL6(RA+2)     .       B - BOXES                         04040
      SLJ(1AA)         ZRO(0)         . EXIT FROM S.R. RLHMERG                  04050
*     FINAL EXIT                                                                04060
1BL   LDA(KA2)         SUB(VA+7)      . DETERMINE NR. OF NULL CELLS             04070
      STA(VA+7)        LIL6(VA+7)     . SET B6=NR. OF NULL CELLS                04080
      ENI1(0)          ENI(0)         . SET B1=0                                04090
      LDA6(BBIN)       ENI(0)         . RELOCATE VALID CELLS                    04100
      STA1(BBIN)       INI1(1)        .    TO LOWER STORAGE AT 30,000B          04110
      ISK6(17777B)     SLJ(1BL+3)     . WHEN COMPLETE, JUMP TO RESTORE          04120
      LDA(VA)          SLJ(1BK+1)     .    SECTION OF (EXIT)                    04130
*     RRUNOUT                                                                   04140
1BM   LDA(VA+6)        INA2(0)        . GENERATE ADDRESS OF KEYR                04150
      SAU(2BJ)         SLJ4(13J)      . SET ADDR KEYR IN (STORE)RJ(STORE)       04160
      INI2(17B)        ENI(0)         . INCREASE B2, 17B                        04170
4BM   ISK2(0)          SLJ(1BM)       . IF B2=ORDSIZE-1,ZERO B2                 04180
      SLJ(1BH)         ENI(0)         . IF NOT, B2=B2+1, GO TO NEW PAIR         04190
*     LRUNOUT                                                                   04200
1BN   LDA(VA+5)        INA1(0)        . GENERATE ADDRESS OF KEYL                04210
      SAU(2BJ)         SLJ4(1BJ)      . SET ADDR OF KEYL IN (STORE),            04220
*                                       RETURN JUMP TO (STORE)                 04230
      INI1(17B)        ENI(0)         . INCREASE INDEX 1 BY 17B                 04240
4BN   ISK1(0)          SLJ(1BN)       . IF B1=ORDSIZE-1, ZERO B1,EXIT           04250
      SLJ(1BH)         ZRO(0)         . IF NOT, B1 = B1 + 1 GOTO NEW PAIR       04260
      END                                                                       04270
                                                                                04280
                                                                                04290
                                                                                04300
      SUBROUTINE HEXSORT(NFLAG1,NFLAG2,NFLAG3,NFLAG4)                           04310
      LOC(ABIN=17000,BBIN=37000,CBIN=57000)                                     04320
      CON(TWENTY=20B)                                                           04330
      SIU1(STOR1)    SIL2(STOR1)     .SAVE                                      04340
      SIU3(STOR2)    SIL5(STOR2)     .    B - BOXES                             04350
      SIU6(STOR3)    ENI1(1)         .       SET B-BOXES                        04360
      ENI2(0)        ENI3(0)         .          TO ZERO                         04370
      ENI5(0)        ENI6(1)         .                                          04380
      ENA(20000B)    SAU(2MERG)      .PRIME SUBROUTINE                          04390
```

```
        LDA(NFLAG2)     SAU(1C)          .                                  04400
        SAU(1TEST)      ENI(0)           .                                  04410
        SLJ4(1RD)       ENI(0)           .                                  04420
        LDA(NFLAG1)     AJP3(1A)         .INSPECT IF LAST BLOCK             04430
        MUI(TWENTY)     SAU(2MERG)       .SET LAST BLOCK PARAMETER          04440
    1A  INI1(-1)        ENI(0)           .                                  04450
        LDA1(ABIN)      SUB2(BBIN)       .MERG ONE KEY                      04460
        AJP(1ASML)      ENI(0)           .                                  04470
        AJP2(1BSML)     SLJ(1ASML)       .    AND TRAILER                   04480
2MERG   ISK2(20000B)    SLJ(1CONT)       .IS BBIN EMPTY                     04490
        ISK1(20000B)    SLJ4(1DUMP)      .IS ABIN EMPTY                     04500
    1C  ISK6(N)         SLJ(1PROC)       .IS READ TAPE EMPTY                04510
        SLJ(2TEST)      ENI(0)           .SERVICE CBIN PRIOR TO EXIT        04520
1PROC   SLJ4(1RD)       ENI(0)           .PROCESS OUT                       04530
        ENI1(1)         SLJ4(1DUMP)      .    REST OF                       04540
        SLJ(1C)         ENI(0)           .       READ TAPE                  04550
1CONT   INI2(-1)        ENI(0)           .                                  04560
        ISK1(20000B)    SLJ(1A)          .IS ABIN EMPTY                     04570
1TEST   ISK6(N)         SLJ(1READ)       .IS READ TAPE EMPTY                04580
        SLJ4(2DUMP)     ZRO(0)           .                                  04590
2TEST   LDA(NFLAG1)     AJP3(1END)       .IS BBIN LAST BLOCK                04600
        LDA(NFLAG1)     INA(-1000B)      .                                  04610
        AJP(1END)       SLJ4(1WRT)       .WAS LAST BLOCK,FULL SIZE          04620
        SLJ(1END)       ENI(0)           .                                  04630
20TCB   SLJ(N)          ENI(0)           .CBIN STORAGE                      04640
        LDA5(N)         STA3(CBIN)       .    ROUTINE                       04650
        INI3(1)         ENI(0)           .       AND AUTOMATIC              04660
        ISK5(17B)       SLJ(20TCB+1)     .          WRITE-OUT               04670
        INI3(-1)        ENI(0)           .             WHEN FULL            04680
        ISK3(17777B)    SLJ(20TCB)       .                                  04690
        SLJ4(1WRT)      ENI(0)           .                                  04700
        SLJ(20TCB)      ENI(0)           .                                  04710
  1RD   SLJ(N)          ENI(0)           .READ ROUTINE                      04720
        CALL READ (NFLAG3,170003,37000B,1)                                  04730
        SLJ(1RD)        ENI(0)           .                                  04740
 1WRT   SLJ(N)          ENI(0)           .WRITE ROUTINE                     04750
        CALL WRITE(NFLAG4,570003,77000B,1)                                  04760
        SLJ(1WRT)       ENI(0)           .                                  04770
1ASML   ENA1(ABIN)      SAU(20TCB+1)     .TRANSFER KEY                      04780
        SLJ4(20TCB)     ENI(0)           .    FROM ABIN                     04790
        INI1(20B)       SLJ(2MERG)       .       TO CBIN                    04800
1BSML   ENA2(BBIN)      SAU(20TCB+1)     .TRANSFER KEY                      04810
        SLJ4(20TCB)     ENI(0)           .    FROM BBIN                     04820
        INI2(20B)       SLJ(2MERG)       .                                  04830
1DUMP   SLJ(N)          INI1(-1)         .TRANSFER                          04840
        ENA1(ABIN)      SAU(20TCB+1)     .    REMAINING                     04850
        SLJ4(20TCB)     ENI(0)           .       KEYS FROM                  04860
        INI1(17B)       ENI(0)           .          ABIN TO                 04870
        ISK1(17777B)    SLJ(1DUMP+1)     .             CBIN                 04880
```

```
      SLJ(1DUMP)      ENI(0)          .                                  04890
2DUMP SLJ(N)          ENI(0)          .TRANSFER                          04900
      ENA2(BBIN)      SAU(20TCB+1)    .   REMAINING                      04910
      SLJ4(20TCB)     ZRO(0)          .      KEYS FROM                   04920
      INI2(17B)       ENI(0)          .         BBIN TO                  04930
      ISK2(17777B)    SLJ(2DUMP+1)    .            CBIN                  04940
      SLJ(2DUMP)      ZRO(0)          .                                  04950
1READ SLJ4(1RD)       ENI(0)          .NON-RETURN JUMP                   04960
      ENI1(1)         SLJ(1A)         .   READ ROUTINE                   04970
 1END LIU1(STOR1)     LIL2(STOR1)     .                                  04980
      LIU3(STOR2)     LIL5(STOR2)     .                                  04990
      LIU6(STOR3)     ENI(0)          .                                  05000
      END                                                                05010
                                                                         05020
                                                                         05030
                                                                         05040
      SUBROUTINE KREMKEY(NFLAG1,NFLAG2,NFLAG3,LISTOUT)                   05050
      REWIND LISTOUT                                                     05060
      REWIND NFLAG3                                                      05070
      NFLAG22=NFLAG2                                                     05080
   50 NFLAG22=NFLAG22-1                                                  05090
      IF(NFLAG22) 200,200,100                                            05100
  100 CALL READ(NFLAG3,17000B,37000B,1)                                  05110
      DO 10 I = 1,512                                                    05120
      INIT = 16761B + 20B*I                                              05130
      NTERM = 17000B + 20B * I                                           05140
      CALL WRITE(LISTOUT,INIT,NTERM,2)                                   05150
   10 CONTINUE                                                           05160
      GO TO 50                                                           05170
  200 IF(NFLAG1) 20,20,201                                               05180
  201 CALL READ(NFLAG3,17000B,37000B,1)                                  05190
      DO 20 I = 1,NFLAG1                                                 05200
      INIT = 16761B+20B*I                                                05210
      NTERM = 17000B+20B*I                                               05220
      CALL WRITE(LISTOUT,INIT,NTERM,2)                                   05230
   20 CONTINUE                                                           05240
      REWIND NFLAG3                                                      05250
      END FILE LISTOUT                                                   05260
      END                                                                05270
                                                                         05280
                                                                         05290
      MACHINE READ(1ARG,2ARG,3ARG,4ARG)                                  05300
                                                                         05310
                                                                         05320
* NOTE  G = ADDRESS OF ENTRY TO TYPEWRITER SENSE ROUTINE                 05330
*       P = ADDRESS OF PROGRAM CONTROL ROUTINE JUMP  ENTRY               05340
* THIS ROUTINE IS FOR PROGRAM CALL                                       05350
                                                                         05360
      LOC(G=31,   P=10)                                                  05370
```

```
* TYPEWRITER MESSAGE CODE TABLE

      CON(T6 =4504041220302204B,   T7 =1120061301050420B,
     *    T8 =1212031200000000B,   T9 =4504041220302204B,
     *    T10=1530121401250420B,   T11=1212031200000000B)
      CON(T12=4504041220302204B,   T13=2006220403260426B,
     *    T14=1411200000000000B,   T15=4504043112140120B,
     *    T40=4504040130152004B,   T41=3406140104060301B,
     *    T42=0414062214163001B,   T43=2022420000000008)

* TAPE UNIT ASSIGNMENT TABLE
*     ASSIGNMENTS FOR CHANNEL 3/4 LIBRARY TAPE

      CON(K0 =0       ,  K1 =332010B,  K2 =332020B,
     *    K3 =332030B,   K4 =332040B,  K5 =552010B,
     *    K6 =552020B,   K7 =552030B,  K8 =552040B)

* ASSORTED CONSTANTS

      CON(R4 =7373737373737373B,  R16=0070000000000000B)

* TAPE READING ROUTINE

1X    SLJ(N)           SLJ(L+5)        . EXIT/ENTRY
1ARG  ZRO(0)           ZRO(0)          . TAPE UNIT NUMBER ARG.
2ARG  ZRO(0)           ZRO(0)          . INITIAL ADDRESS
3ARG  ZRO(0)           ZRO(0)          . TERMINAL ADDRESS
4ARG  ZRO(0)           ZRO(0)          . MODE, 1 = BINARY, 2 = BCD
      LDA7(1ARG)       INA(-11B)       .
      SIU1(19X)        AJP3(L+3)       . CHECK FOR TU ASSIGNMENT
      ENA(T40)         SLJ4( 3)        . GO PRINT ERROR
      SLJ( P)          ZRO(0)          .
      INA(11B)         STA(K0)         .
      LIL1(K0)         LDA1(K0)        .
      ADD7(4ARG)       SAU(4X)         . SET TU CONTROL CODES
      SCL(777B)        SAL(4X)         . INITIALIZE
      SAL(5X)          SAL(11X)        .
      INA(3)           SAU(8X)         .
      INA(2)           SAU(9X)         .
      INA(1)           SAU(11X)        .
      INA(1)           SAU(7X)         .
      LRS(18)          ENA(0)          .
      LLS(3)           SAL(2X)         .
      SAL(3X)          SAU(12X)        .
```

```
        LLS(39)          LDQ(R16)        .                                  05870
        STA(KO)          LDA(5X)         .                                  05880
        SSU(KO)          STA(5X)         .                                  05890
        LDA7(2ARG)       SAU(5X)         .                                  05900
        LDA(6X)          SSU(KO)         .                                  05910
        STA(6X)          ENI(O)          .                                  05920
2X      LDA7(3ARG)       SAL(N)          .                                  05930
        SAU(6X)          SIL2(19X)       .                                  05940
3X      ENI2(4)          LIL1(N)         .                                  05950
4X      EXF(N)           EXF7(N)         . SELECT UNIT                      05960
5X      EXF(N)           EXF7(N)         . ACTIVATE CHANNEL                 05970
6X      EXF(N)           LDA7(2ARG)      .                                  05980
        SUB(R4)          LLS(48)         .                                  05990
        INI1(-1)         LDA1(Z)         .                                  06000
        INI1(1)          SUB(R4)         .                                  06010
7X      EXF7(N)          SLJ(15X)        . SENSE ENDFILE                    06020
8X      EXF7(N)          SLJ(L+4)        . SENSE PARITY ERROR               06030
9X      EXF7(N)          SLJ(10X)        . SENSE LENGTH ERROR               06040
        QJP1(19X)        AJP(3X)         . SENSE FOR BAD RECORD             06050
        SLJ(19X)         ZRO(O)          . EXIT                             06060
        IJP2(L+2)        SLJ(14X)        . SENSE FIVE PARITY ERRORS         06070
10X     IJP2(L+1)        SLJ(13X)        . SENSE FIVE LENGTH ERRORS         06080
        AJP(3X)          QJP(3X)         . SENSE FOR BAD RECORD             06090
11X     EXF(N)           EXF7(N)         . BACKSPACE                        06100
12X     LIL1(N)          SLJ(4X)         . RETURN TO REREAD                 06110
13X     ENA(T6)          SLJ(17X)        . SET TO INDICATE TROUBLE          06120
14X     ENA(T9)          SLJ(17X)        .                                  06130
15X     ENA(T12)         SLJ(18X)        .                                  06140
17X     SLJ4( G)         ZRO(O)          . TROUBLE EXIT                     06150
        AJP(19X)         AJP3(3X)        . SENSE ACTION TO TAKE             06160
        SLJ(11X)         ZRO(O)          .                                  06170
18X     SLJ4( G)         ZRO(O)          . TROUBLE EXIT                     06180
        AJP(3X)          AJP3(3X)        . SENSE ACTION TO TAKE             06190
        SLJ(11X)         ZRO(O)          .                                  06200
19X     ENI1(N)          ENI2(N)         . RE-STORE INDEXES                 06210
        SLJ(1X)          ZRO(O)          .                                  06220
        END                                                                 06230
                                                                            06240
                                                                            06250
      MACHINE WRITE(1ARG,2ARG,3ARG,4ARG)                                    06260
                                                                            06270
                                                                            06280
* NOTE  G = ADDRESS OF ENTRY TO TYPEWRITER SENSE ROUTINE                    06290
*       P = ADDRESS OF PROGRAM CONTROL ROUTINE JUMP  ENTRY                  06300
* THIS ROUTINE IS FOR PROGRAM CALL                                          06310
                                                                            06320
      RSV(T=2)                                                              06330
      LOC(G=31,  P=10)                                                      06340
                                                                            06350
```

```
* TYPEWRITER MESSAGE CODE TABLE

      CON(T14=14112000000000003B,   T15=4504043112140120B,
     *    T16=04112006130105048,   T17=2012120312000000B,
     *    T18=4504043112140120B,   T19=0415301214012504B,
     *    T20=2012120312000000B,   T21=4504043112140120B,
     *    T22=0420062204032604B,   T23=0130152000000000B,
     *    T40=4504040130152004B,   T41=3406140104060301B,
     *    T42=0414062214163001B,   T43=2022420000000000B)

* TAPE UNIT ASSIGNMENT TABLE
*     ASSIGNMENTS FOR CHANNEL 3/4 LIBRARY TAPE

      CON(K0 =0        ,  K1 =442010B,  K2 =442020B,
     *    K3 =442030B,   K4 =442040B,  K5 =662010B,
     *    K6 =662020B,   K7 =662030B,  K8 =662040B)

* ASSORTED CONSTANTS

      CON(R4 =7373737373737373B,  R16=0070000000000000B)

* TAPE WRITING ROUTINE

1Y    SLJ(N)           SLJ(L+5)        . EXIT/ENTRY
1ARG  ZRO(0)           ZRO(0)          . TAPE UNIT NUMBER ARG.
2ARG  ZRO(0)           ZRO(0)          . INITIAL ADDRESS
3ARG  ZRO(0)           ZRO(0)          . TERMINAL ADDRESS
4ARG  ZRO(0)           ZRO(0)          . MODE, 1 = BINARY, 2 = BCD
      LDA7(1ARG)       INA(-11B)       .
      SIU1(21Y)        AJP3(L+3)       . CHECK FOR TU ASSIGNMENT
      SLJ( P)          ZRO(0)          .
      ENA(T40)         SLJ4( G)        . GO PRINT ERROR
      INA(11B)         STA(K0)         .
      LIL1(K0)         LDA1(K0)        .
      ADD7(4ARG)       SAU(2Y)         . SET TU CONTROL CODES
      SAU(9Y)          SCL(777B)       . INITIALIZE
      SAL(2Y)          SAL(4Y)         .
      SAL(8Y)          SAL(10Y)        .
      INA(3)           SAU(5Y)         .
      SAU(11Y)         INA(2)          .
      SAU(6Y)          SAU(12Y)        .
      INA(1)           SAU(8Y)         .
      SAU(7Y)          SAU(13Y)        .
      LRS(15)          SAL(1YB)        .
      ENI(0)           ENQ(0)          .
```

```
        LLS(39)          LDQ(R16)        .                                   06850
        STA(K0)          LDA(4Y)         .                                   06860
        SSU(K0)          STA(4Y)         . INSERT CHANNEL SELECTION          06870
        LDA(10Y)         SSU(K0)         .                                   06880
        STA(10Y)         SIL2(21Y)       .                                   06890
1YB     LDA7(3ARG)       SAL(N)          .                                   06900
        INA(-1)          SAU(14YB)       .                                   06910
        SAU(15YB)        SAL(20YB)       .                                   06920
        LDA7(2ARG)       SAU(4Y)         .                                   06930
        SAU(10Y)         SAU(14YA)       .                                   06940
        SAL(15YA)        SAL(20YA)       .                                   06950
2Y      EXF(N)           EXF7(N)         . SELECT UNIT                       06960
4Y      EXF(N)           EXF7(N)         . ACTIVATE CHANNEL                  06970
5Y      EXF7(N)          SLJ(17Y)        . SENSE PARITY ERROR                06980
6Y      EXF7(N)          SLJ(19Y)        . SENSE LENGTH ERROR                06990
7Y      EXF7(N)          SLJ(21Y)        . SENSE END OF TAPE                 07000
        SLJ(16Y)         ZRO(0)          .                                   07010
8Y      EXF(N)           EXF7(N)         . BACKSPACE                         07020
9Y      EXF(N)           SLJ(13YA)       . SELECT UNIT                       06690
10Y     EXF(N)           EXF7(N)         . WRITE BAD RECORD INDICATION       07040
11Y     EXF7(N)          SLJ(20YA)       . SENSE PARITY ERROR                07050
12Y     EXF7(N)          SLJ(20YA)       . SENSE LENGTH ERROR                07060
13Y     EXF7(N)          SLJ(20YA)       . SENSE END OF TAPE                 07070
        SLJ(16Y)         ZRO(0)          .                                   07080
13YA    IJP2(L+1)        SLJ(2Y)         .                                   06790
14YA    LDA(N)           STA(T)          . SET BAD RECORD INDICATION         07090
14YB    LDA(N)           STA(T+1)        .                                   07100
15YA    LDA(R4)          STA(N)          .                                   07110
15YB    STA(N)           SLJ(10Y)        .                                   06780
16Y     ENA(T21)         SLJ4( G)        . TROUBLE EXIT                      07140
        AJP(2Y)          ENI2(0)         . SENSE ACTION TO TAKE              07150
        AJP3(21Y)        SLJ(8Y)         .                                   07160
17Y     ENA(T18)         SLJ4( G)        . TROUBLE EXIT                      07170
18Y     AJP3(21Y)        ENI2(1)         . SENSE ACTION TO TAKE              07180
        AJP(8Y)          ENI2(0)         .                                   07190
        SLJ(8Y)          ZRO(0)          .                                   07200
19Y     ENA(T15)         SLJ4( G)        . TROUBLE EXIT                      07210
        SLJ(18Y)         ZRO(0)          .                                   07220
20YA    LDA(T)           STA(N)          . RESET INIT AND FINAL CELLS        07230
20YB    LDA(N)           STA(N)          .                                   07240
        SLJ(2Y)          ZRO(0)          .                                   07250
21Y     ENI1(N)          ENI2(N)         . RE-STORE INDEXES                  07260
        SLJ(1Y)          ZRO(0)          .                                   07270
        END                                                                  07280
        END                                                                  07290
```

```
1

 * FORTSAT * FORTSAT * FORTSAT * FORTSAT * FORTSAT * FORTSAT * FORTSAT *

 *      USNPS FORTRAN SYSTEM RESIDENT PROGRAM  25 SEPT 1962

 *           FORTSAT IS A  5K BIAS VERSION OF FORTRAN SIXTY
 *           WITH USNPGS SATELLITE SYSTEM IMPLEMENTATION

 *           ( CDC VERSION 15 MARCH 1962  - - MODIFIED )
 *             LAST UPDATE - - 20 APRIL 1963
 *               R.L. HOGG / D. C. GLOVER


 *           I/O DESIGNATORS

 *                T - KEYBOARD OR TYPEWRITER (01)
 *                F - FLEX PAPER TAPE (02)
 *                P - BCD PAPER TAPE (INPUT) (03)
 *                C - 521 READER (10) PUNCH (12)
 *                1 - TU1 CH 3-4 (21)
 *                2 - TU2 CH 3-4 (22)
 *                3 - TU3 CH 3-4 (23)
 *                4 - TU4 CH 3-4 (24)
 *                5 - TU5 CH 5-6 (31)
 *                6 - TU6 CH 5-6 (32)
 *                7 - TU7 CH 5-6 (33)
 *                8 - TU8 CH 5-6 (34)
 *                9 - TYPEWRITER (01)
 *                P - TU4 (OUTPUT) (24)
 *                M - MEMORY (06)

 *           RESIDENT ROUTINES (NO CALL REQUIRED)

 *                CALL -        (TU NO.) (UP TO EIGHT PROGRAM NAMES)
 *                CONTROL -     (CONTROL MEDIUM) (CONTROL STATEMENT OUTPUT
 *                              MEDIUM) (RUNNING PROGRAM ERROR MEDIUM)
 *                REWIND -      (TU NO.)
 *                BKSP -        (TU NO.)
 *                TIME -        (NO ARGS)  TYPES OUT TIME
 *                LIMIT -       (MINUTES) (SECONDS)
 *                CLOCK -       (NO ARGS)  STARTS CLOCK
 *                EFMARK -      (TU NO.)  THIS IS 17 CODE
 *                RUN -         (NO ARGS)  EXECUTES LAST PROGRAM CALLED
 *                HOLD -        (NO ARGS) EFFECTIVE FOR NEXT CALL ONLY


APPENDIX  - IV -    PAGE    1                          HOGG/GLOVER  1963
```

```
*                CLEAR -       (NO ARGS)  CLEARS MEMORY FROM
*                              RESIDENT BIAS TO CELL 77776
*                SWITCH -      (UP TO EIGHT SETTINGS) NOTE - FIRST THREE
*                              ARE ORDED WITH PHYSICAL SETTING
*                              OF CONSOLE KEYS
*                INPUT -       (MNEMONIC EQPT CODE) (BIOCTAL EQPT CODE)
*                OUTPUT -      (MNEMONIC EQPT CODE) (BIOCTAL EQPT CODE)
*                STOPCK -      (NO ARGS)
*                SET -         (STARTING ADDRESS M-FILE) (RECORD SIZE
*                                                        IN WORDS)
*                SKIP -        (MEDIUM)  (I.D. CODE)
*                LOCK -        (TU NO.)
*                MONITOR -     (INPUT) (OUTPUT) (MODE) (TIME LIMIT)
*                              (ALT OUTPUT) (SOURCE LISTING) (MAP LISTING)
*                SR -          (NO ARGS)  THIS IS A SERVICE ROUTINE CALL
*                              WHICH CALLS THE FOLLOWING  LIB ROUTINES,
*                              COPYS, TRANSFER, VFYLIB, LIST,
*                              TROUBLE, READMT, WRITEMT, LIBFIX

*          BOOTSTRAP FROM CHANNEL 3-4
*          CURRENT LIBRARY LISTING FOLLOWS RESIDENT LISTING



S1     MACHINE RESIDENT


       RSV(T=11, BUF65=9, SATBUF=1)
       LOC(Z=0,J=26,C=60,R=100,W=300,M=531,S=5000)
*      TO CHANGE RESIDENT BIAS, SET S=NEW VALUE

* BCD CODE TABLE

       CON(F0 =12B,F1 =01B,F2 =02B,F3 =03B,F4 =04B,F5 =05B,F6 =06B,
      *    F7 =07B,F10=10B,F11=11B,F12=61B,F13=62B,F14=63B,F15=64B,
      *    F16=65B,F17=66B,F20=67B,F21=70B,F22=71B,F23=41B,F24=42B,
      *    F25=43B,F26=44B,F27=45B,F30=46B,F31=47B,F32=50B,F33=51B,
      *    F34=22B,F35=23B,F36=24B,F37=25B,F40=26B,F41=27B,F42=30B,
      *    F43=31B,F44=73B,F45=53B,F46=34B,F47=13B,F50=54B,F51=21B,
      *    F52=60B,F53=40B,F54=74B,F55=33B,F56=20B,F57=14B          )

* GENERALIZED SUBSTIIUTION TABLE

*          INDEX ON GO, INDEX = CHARACTER BCD CODE
*          UPPER SIX BITS ARE ANALEX PRINTER FIRING ORDER
```

```
*          NEXT 12 BITS ARE CARD IMAGE ALGORITHM CONTROL
*          NEXT 18 BITS ARE NOT USED
*          SEVENTH CHARACTER IS INPUT EQUIP. CONTROL CODE
*          SEVENTH CHARACTER IS OUTPUT EQUIP. CONTROL CODE

       CON(G0 =6100000000000000B,  G1 =0100100000002177B,
      *    G2 =0200200000002222B,  G3 =0300400000002323B,
      *    G4 =0401000000002424B,  G5 =0502000000003131B,
      *    G6 =0604000000003232B,  G7 =0710000000003333B,
      *    G10=1020000000003434B,  G11=1140000000000101B,
      *    G12=0000040000007777B,  G13=4720400000007777B,
      *    G14=4521000000007777B,  G15=6100000000007777B,
      *    G16=6100000000007777B,  G17=6100000000007777B)
       CON(G20=6300000000007777B,  G21=5200140000007777B,
      *    G22=3400240000007777B,  G23=3500440000000101B,
      *    G24=3601040000007777B,  G25=3702040000007777B,
      *    G26=4004040000007777B,  G27=4110040000000000B,
      *    G30=4220040000007777B,  G31=4340040000007777B,
      *    G32=6100000000007777B,  G33=5420440000007777B,
      *    G34=5021040000007777B,  G35=6100000000007777B,
      *    G36=6100000000007777B,  G37=6100000000007777B)
       CON(G40=4500020000007777B,  G41=2300120000007777B,
      *    G42=2400220000007777B,  G43=2500420000007777B,
      *    G44=2601020000000606B,  G45=2702020000007777B,
      *    G46=3004020000007777B,  G47=3110020000000324B,
      *    G50=3220020000007777B,  G51=3340020000007777B,
      *    G52=6100000000007777B,  G53=6220420000007777B,
      *    G54=5321020000007777B,  G55=6100000000007777B,
      *    G56=6100000000007777B,  G57=6100000000007777B)
       CON(G60=4600010000007777B,  G61=1200110000007777B,
      *    G62=1300210000007777B,  G63=1400410000010128,
      *    G64=1501010000007777B,  G65=1602010000007777B,
      *    G66=1704010000002028,   G67=2010010000007777B,
      *    G70=2120010000007777B,  G71=2240010000007777B,
      *    G72=6100000000007777B,  G73=4420410000007777B,
      *    G74=5121010000007777B,  G75=6100000000007777B,
      *    G76=6100000000007777B,  G77=6100000000007777B)

* CHARACTER CONVERSION TABLE

*          INDEX = CHARACTER CODE
*          UPPER SIX BITS ARE TYPEWRITER CASE CONTROL
*                77 = UPPER CASE
*                00 = LOWER CASE
*          2ND CHAR. = TYPE OUTPUT CHAR. CONVERSION
*          3RD CHAR. = FLEX CASE CONTROL
*          4TH CHAR. = FLEX OUTPUT CHAR. CONVERSION
*          5TH CHAR. = TYPE UPPER CASE INPUT CONVERSION
```

```
*          6TH CHAR. = TYPE LOWER CASE INPUT CONVERSION
*          7TH CHAR. = FLEX UPPER CASE INPUT CONVERSION
*          8TH CHAR. = FLEX LOWER CASE INPUT CONVERSION

      CON(K0 =7746775620202020B,   K1 =0074007423232323B,
     *    K2 =0070007020132020B,   K3 =0064006446464646B,
     *    K4 =0062006220202020B,   K5 =0066006670707070B,
     *    K6 =0072007245454545B,   K7 =0060006044444444B,
     *    K10=0033003320202020B,   K11=0037003743434343B,
     *    K12=0056005651515151B,   K13=0002774267676767B,
     *    K14=0052005271717171B,   K15=7746775647474747B,
     *    K16=7746775663636363B,   K17=7746775625252525B)
      CON(K20=0004000465656565B,   K21=0044004431313131B,
     *    K22=0024002464646464B,   K23=0001000162626262B,
     *    K24=0034003422222222B,   K25=0017001730303030B,
     *    K26=0031003166666666B,   K27=0027002727272727B,
     *    K30=0025002561616161B,   K31=0021002126262626B,
     *    K32=7746775641414141B,   K33=0040004620101010B,
     *    K34=7737775424242424B,   K35=7746775650505050B,
     *    K36=7746775642424242B,   K37=7746775634111111B)
      CON(K40=0052005233332020B,   K41=0032003220202020B,
     *    K42=0036003673731373B,   K43=0011001120202020B,
     *    K44=0007000720215421B,   K45=0006000620202020B,
     *    K46=0003000320606033B,   K47=0015001520202020B,
     *    K50=0035003520205320B,   K51=0012001220202020B,
     *    K52=7746775620404040B,   K53=7762775020202020B,
     *    K54=7774774420203474B,   K55=7776775620202020B,
     *    K56=7746775674121212B,   K57=7746775620202020B)
      CON(K60=0046774620070707B,   K61=0030003020202020B,
     *    K62=0023002353040404B,   K63=0016001620202020B,
     *    K64=0022002220030303B,   K65=0020002020202020B,
     *    K66=0026002620050505B,   K67=0013001320202020B,
     *    K70=0005000520020202B,   K71=0014001420202020B,
     *    K72=7746775620060606B,   K73=0042004220202020B,
     *    K74=7756005454010101B,   K75=7746775620202020B,
     *    K76=7746775620202020B,   K77=7746775620202020B)

* PROGRAM NAME TABLE

      CON(P0 =4446457123465100B,   P1 =6565654564000000B,
     *    P2 =7145472423000000B,   P3 =4624234724230000B,
     *    P4 =6343466342000000B,   P5 =2223464763420000B,
     *    P6 =6346452351464300B,   P7 =2251000000000000B,
     *    P10=6361434300000000B,   P11=2226712363700000B,
     *    P12=5124450000000000B,   P13=7046436400000000B,
     *    P14=6343656151000000B,   P15=2242714700000000B,
     *    P16=2265230000000000B,   P17=2371446500000000B)
      CON(P20=4371447123000000B,   P21=5165267145640000B,
```

```
*     P22=4346634200000000B,   P23=6566446151420000B,
*     P24=6242224700000000B,   P25=0000000000000000B,
*     P26=0000000000000000B,   P27=0000000000000000B,
*     P30=0000000000000000B,   P31=0000000000000000B,
*     P32=0000000000000000B,   P33=0000000000000000B,
*     P34=0000000000000000B,   P35=0000000000000000B,
*     P36=0000000000000000B,   P37=0000000000000000B)

* BCD MESSAGE CODE TABLE

      CON(E0 =4751466751614420B,   E1 =4546232063614343B,
*     E2 =6564000000000000B,   E3 =2223465161676520B,
*     E4 =4371447123000000B,   E5 =4546232061256171B,
*     E6 =4361624365404040B,   E7 =0000000000000000B,
*     E8 =0000000000000000B,   E9 =2445612222716745B,
*     E10=6564207145472423B,   E11=2044656471244400B,
*     E12=2445612222716745B,   E13=6564204624234724B)
      CON(E14=2320446564712444B,   E15=0000000000000000B,
*     E16=2371446520437144B,   E17=7123000000000000B,
*     E18=2346462044614530B,   E19=2061516724446545B,
*     E20=2322000000000000B,   E21=0000000000000000B,
*     E22=2044714524236522B,   E23=0000000000000000B,
*     E24=2022656346456422B,   E25=0000000000000000B)

* TYPEWRITER MESSAGE CODE TABLE

      CON(T0 =4504041605201636B,   T1 =0424340704201212B,
*     T2 =0312000000000000B,   T3 =4504041630122204B,
*     T4 =1220302204201212B,   T5 =0312000000000000B,
*     T6 =4504041220302204B,   T7 =1120061301050420B,
*     T8 =1212031200000000B,   T9 =4504041220302204B,
*     T10=1530121401250420B,   T11=1212031200000000B)
      CON(T12=4504041220302204B,   T13=2006220403260426B,
*     T14=1411200000000000B,   T15=4504043112140120B,
*     T16=0411200613010504B,   T17=2012120312000000B,
*     T18=4504043112140120B,   T19=0415301214012504B,
*     T20=2012120312000000B,   T21=4504043112140120B,
*     T22=0420062204032604B,   T23=0130152000000000B)

* SERVICE ROUTINES CALLED BY SR

      CON(COPYS=6346473022000000B, TRNSFER=2351614522666551B,
*     VFYLIB=2566304371620000B, LIST=4371222300000000B,
*     TROUBLE=2351462462436500B, READMT=5165616444230000B,
*     WRITEMT=2651712365442300B, LIBFIX=4371626671270000B)
```

```
* 160 CALL ROUTINE CONSTANTS

      CON(IDENT=0000000001231111B). RS-022 IDENT
      CON(IDEN2=0000000001231122B). OSAP IDENT
      CON(IDEN3=0000000001231133B). OSAS IDENT
      CON(IDEN4=0000000001231144B). OSAS LOADER, A4.01 IDENT
      CON(IDEN5=0000000001231155B). OSAP LOADER
      CON(E=0, E=0, E=0, E=0)
      CON(SIZE=77B, SIZE2=773B, SIZE3=1214B, SIZE4=77B, SIZE5=23B)
      CON(E=0, E=0, E=0, E=0)
      CON(TAPE1=0000000000332011B)
      CON(BLOCKS=700B). LIMIT ON 160 LIB CALL BLOCK READ BEFORE EXIT

* TAPE CHANNEL ASSIGNMENT TABLE

      CON(U0 =0033200000442000B,  U1 =0055200000662000B)

* ASSORTED CONSTANTS

      CON(R0 =01,R1 =10,R2 =100,R3 =66465123516145408,
     *    R4 =7373737373737373B,R5 =4500000000000000B,
     *    R6 =2020202020202020B,R7 =7760000000000000B,
     *    R10=6545646671436500B,R11=0000000100000000B,
     *    R12=3777777777777704B,R13=0351244513000000B,
     *    R14=0041254400000000B,R15=6646512362714500B,
     *    R16=0070000000000000B,R17=5165446151420000B)

* DD65 PRINTER SIMULATOR CONSTANTS

      CON(DDSWT=5252525252525252B)
      CON(ND65R=1000B)

*          PSUEDO PRINTER DRUM DESIGNATOR TABLE

      CON(ITAB=3400000006001400B,
     1    LTAB=3400000006241420B,
     2    LTAB=3400000006501440B,
     3    LTAB=3400000006741460B,
     4    LTAB=3400000007201500B,
     5    LTAB=3400000007441520B,
     6    LTAB=3400000007701540B,
     7    LTAB=3400000010141560B)
      CON(JTAB=3400000010401600B,
     1    LTAB=3400000010641620B,
     2    LTAB=3400000011101640B,
```

```
3     LTAB=340000001134160B,
4     LTAB=340000001160170OB,
5     LTAB=340000001204172OB,
6     LTAB=340000001230174OB,
7     LTAB=340000001254176OB)
  CON(KTAB=340000001300100OB,
1     LTAB=340000001324102OB,
2     LTAB=340000001350104OB,
3     LTAB=340000001374106OB,
4     LTAB=340000001420110OB,
5     LTAB=340000001444112OB,
6     LTAB=340000001470114OB,
7     LTAB=340000001514116OB)
  CON(MTAB=340000001540120OB,
1     LTAB=340000001564122OB,
2     LTAB=340000001610124OB,
3     LTAB=340000001634126OB,
4     LTAB=340000001660130OB,
5     LTAB=340000001704132OB,
6     LTAB=340000001730134OB,
7     LTAB=340000001754136OB)
```

* INITIAL START/EXITS AND ENTRIES TO RESIDENT ROUTINES

```
SLJ(N)           SLJ(L)          .                                  00000
SLJ(N)           SLJ(1SENS)      . INTERRUPT EXIT/ENTRY             00001
SLJ(Z+40B)       ZRO(0)          . JUMP TO CONTINUE LOAD            00002
SLJ(N)           SLJ(3A)         . BCD READ EXIT/ENTRY              00003
SLJ(N)           SLJ(65DPY)      . BCD WRITE EXIT/ENTRY             00004
SLJ(N)           SLJ(5A)         . BIN READ EXIT/ENTRY              00005
SLJ(N)           SLJ(6A)         . BIN WRITE EXIT/ENTRY             00006
SLJ(N)           SLJ(7A)         . BACKSPACE EXIT/ENTRY             00007
SLJ(N)           SLJ(8A)         . REWIND EXIT/ENTRY                00010
SLJ(N)           SLJ(9A)         . ENDFILE EXIT/ENTRY               00011
EXF(62540B)      SLJ(Z+40B)      . JUMP TO CONTINUE LOAD/RELOAD     00012
SLJ(N)           SLJ(9M)         . NORMAL RETURN TO MONITOR         00013
SLJ(N)           SLJ(10A)        . GO TO CHAIN CONNECTOR            00014
SLJ(N)           SLJ(1E)         . GO INDICATE EQUIPMENT FAILURE    00015
SLJ(N)           SLJ(2E)         . GO INDICATE PROGRAMMING ERROR    00016
SLJ(N)           SLJ(11A)        . BCD MESSAGE EXIT/ENTRY           00017
SLJ(N)           SLJ(N)          . SPARE                            00020
ZRO(0)           ZRO(0Z)         . ADDR OF PROGRAM START TABLE      00021
SLJ(2A)          ZRO(0)          . JUMP TO RESUME MONITOR CONTROL   00022
SLJ(N)           SLJ(5GG)        . TYPE MESSEGE/SENSE FOR ACTION    00023
SLJ(N)           SLJ(N)          . SPARE                            00024
SLJ(N)           SLJ(3VV)        . GO LOCK TAPE                     00025
SLJ(N)           SLJ(N)          . SPARE                            00026
EXF(32011B)      EXF(32005B)     . REWIND LIBRARY TAPE              00027
```

```
      ENI1(0)          EXF7(32000B)      .                                    00030
      EXF3(Z)          EXF7(32000B)      . REREAD FIRST RESIDENT BLOCK        00031
      ENA(1PCON)       SAU(Z+10B)        . SET JUMP TO PROGRAM CONTROL        00032
      ENA(Z+35B)       SAL(Z+20B)        . SET JUMP TO RELOAD RESIDENT        00033
      ENI2(48)         ENI3(12)          .                                    00034
      ENA(R+54)        STA(Z+3)          .                                    00035
      EXF3(R)          EXF7(32000B)      . READ RESIDENT BLOCK                00036
      EXF3(R+54)       ENQ(0)            .                                    00037
      EXF7(32003B)     SLJ(Z+35B)        . SENSE PARITY ERROR                 00040
      EXF7(32005B)     SLJ(Z+35B)        . SENSE LENGTH ERROR                 00041
      LDA1(R+6)        STA2(Z)           . TRANSFER WORD TO RESIDENT AREA     00042
      ISK2(T+11)       SLJ(Z+53B)        . SENSE END OF RESIDENT              00043
      EXF(32005B)      SLJ(Z+55B)        .                                    00044
      ISK1(47)         SLJ(Z+50B)        . SENSE END OF BLOCK                 00045
      SLJ(Z+44B)       ZRO(0)            . GO READ NEXT BLOCK                 00046
      STQ3(Z+40B)      IJP3(Z+55B)       . CLEAR SIMULATION LOCATIONS         00047
      STQ(Z+55B)       STQ(Z+56B)        .                                    00050
      STQ(Z)           SLJ(1PCON)        . EXIT TO PROGRAM CONTROL            00051
      ZRO(0)           ZRO(0)            .                                    00052
      ZRO(0)           ZRO(0)            .                                    00053
      ZRO(0)           ZRO(0)            .                                    00054
      ZRO(0)           ZRO(0)            .                                    00055
      ZRO(0)           ZRO(0)            .                                    00056
      ZRO(0)           ZRO(0)            .                                    00057

* CONTROL INFORMATION AND MONITOR TAPE ASSIGNMENT TABLE

      ZRO(0)           ZRO(0)            . END OF FORTRAN MAP FILE            00060
      ZRO(0)           ZRO(0)            . START OF FORTRAN MAP FILE          00061
      ZRO(0)           ZRO(S)            . INITIAL PROGRAM BIAS               00062
      ZRO(0)           ZRO(S)            . CURRENT PROGRAM BIAS               00063
      ZRO(0)           ZRO(J)            . INITIAL NUMBER OF PROGRAMS         00064
      ZRO(0)           ZRO(J)            . CURRENT NUMBER OF PROGRAMS         00065
      ZRO(0)           ZRO1(?0)          . ADDRESS OF PROGRAM-NAME TABLE      00066
      ZRO(0)           ZRO1(0Z)          . ADDRESS OF PROGRAM-START TABLE     00067
      ZRO(0)           ZRO(1)            . MONITOR TAPE-ASSIGNMENT TABLE      00070
      ZRO(0)           ZRO(2)            .                                    00071
      ZRO(0)           ZRO(3)            .                                    00072
      ZRO(0)           ZRO(4)            .                                    00073
      ZRO(0)           ZRO(5)            .                                    00074
      ZRO(0)           ZRO(6)            .                                    00075
      ZRO(0)           ZRO(7)            .                                    00076
      ZRO(0)           ZRO(8)            .                                    00077

* READ BUFFER

      ZRO(0)           ZRO(0)            . READ BUFFER                        00100
```

```
ZRO(0)          ZRO(0)          .               00101
ZRO(0)          ZRO(0)          .               00102
ZRO(0)          ZRO(0)          .               00103
ZRO(0)          ZRO(0)          .               00104
ZRO(0)          ZRO(0)          .               00105
ZRO(0)          ZRO(0)          .               00106
ZRO(0)          ZRO(0)          .               00107
ZRO(0)          ZRO(0)          .               00110
ZRO(0)          ZRO(0)          .               00111
ZRO(0)          ZRO(0)          .               00112
ZRO(0)          ZRO(0)          .               00113
ZRO(0)          ZRO(0)          .               00114
ZRO(0)          ZRO(0)          .               00115
ZRO(0)          ZRO(0)          .               00116
ZRO(0)          ZRO(0)          .               00117
ZRO(0)          ZRO(0)          .               00120
ZRO(0)          ZRO(0)          .               00121
ZRO(0)          ZRO(0)          .               00122
ZRO(0)          ZRO(0)          .               00123
ZRO(0)          ZRO(0)          .               00124
ZRO(0)          ZRO(0)          .               00125
ZRO(0)          ZRO(0)          .               00126
ZRO(0)          ZRO(0)          .               00127
ZRO(0)          ZRO(0)          .               00130
ZRO(0)          ZRO(0)          .               00131
ZRO(0)          ZRO(0)          .               00132
ZRO(0)          ZRO(0)          .               00133
ZRO(0)          ZRO(0)          .               00134
ZRO(0)          ZRO(0)          .               00135
ZRO(0)          ZRO(0)          .               00136
ZRO(0)          ZRO(0)          .               00137
ZRO(0)          ZRO(0)          .               00140
ZRO(0)          ZRO(0)          .               00141
ZRO(0)          ZRO(0)          .               00142
ZRO(0)          ZRO(0)          .               00143
ZRO(0)          ZRO(0)          .               00144
ZRO(0)          ZRO(0)          .               00145
ZRO(0)          ZRO(0)          .               00146
ZRO(0)          ZRO(0)          .               00147
ZRO(0)          ZRO(0)          .               00150
ZRO(0)          ZRO(0)          .               00151
ZRO(0)          ZRO(0)          .               00152
ZRO(0)          ZRO(0)          .               00153
ZRO(0)          ZRO(0)          .               00154
ZRO(0)          ZRO(0)          .               00155
ZRO(0)          ZRO(0)          .               00156
ZRO(0)          ZRO(0)          .               00157
ZRO(0)          ZRO(0)          .               00160
ZRO(0)          ZRO(0)          .               00161
```

```
ZRO(0)          ZRO(0)          .               00162
ZRO(0)          ZRO(0)          .               00163
ZRO(0)          ZRO(0)          .               00164
ZRO(0)          ZRO(0)          .               00165
ZRO(0)          ZRO(0)          .               00166
ZRO(0)          ZRO(0)          .               00167
ZRO(0)          ZRO(0)          .               00170
ZRO(0)          ZRO(0)          .               00171
ZRO(0)          ZRO(0)          .               00172
ZRO(0)          ZRO(0)          .               00173
ZRO(0)          ZRO(0)          .               00174
ZRO(0)          ZRO(0)          .               00175
ZRO(0)          ZRO(0)          .               00176
ZRO(0)          ZRO(0)          .               00177
ZRO(0)          ZRO(0)          .               00200
ZRO(0)          ZRO(0)          .               00201
ZRO(0)          ZRO(0)          .               00202
ZRO(0)          ZRO(0)          .               00203
ZRO(0)          ZRO(0)          .               00204
ZRO(0)          ZRO(0)          .               00205
ZRO(0)          ZRO(0)          .               00206
ZRO(0)          ZRO(0)          .               00207
ZRO(0)          ZRO(0)          .               00210
ZRO(0)          ZRO(0)          .               00211
ZRO(0)          ZRO(0)          .               00212
ZRO(0)          ZRO(0)          .               00213
ZRO(0)          ZRO(0)          .               00214
ZRO(0)          ZRO(0)          .               00215
ZRO(0)          ZRO(0)          .               00216
ZRO(0)          ZRO(0)          .               00217
ZRO(0)          ZRO(0)          .               00220
ZRO(0)          ZRO(0)          .               00221
ZRO(0)          ZRO(0)          .               00222
ZRO(0)          ZRO(0)          .               00223
ZRO(0)          ZRO(0)          .               00224
ZRO(0)          ZRO(0)          .               00225
ZRO(0)          ZRO(0)          .               00226
ZRO(0)          ZRO(0)          .               00227
ZRO(0)          ZRO(0)          .               00230
ZRO(0)          ZRO(0)          .               00231
ZRO(0)          ZRO(0)          .               00232
ZRO(0)          ZRO(0)          .               00233
ZRO(0)          ZRO(0)          .               00234
ZRO(0)          ZRO(0)          .               00235
ZRO(0)          ZRO(0)          .               00236
ZRO(0)          ZRO(0)          .               00237
ZRO(0)          ZRO(0)          .               00240
ZRO(0)          ZRO(0)          .               00241
ZRO(0)          ZRO(0)          .               00242
```

```
        ZRO(0)          ZRO(0)          .                                   00243
        ZRO(0)          ZRO(0)          .                                   00244
        ZRO(0)          ZRO(0)          .                                   00245
        ZRO(0)          ZRO(0)          .                                   00246
        ZRO(0)          ZRO(0)          .                                   00247
        ZRO(0)          ZRO(0)          .                                   00250
        ZRO(0)          ZRO(0)          .                                   00251
        ZRO(0)          ZRO(0)          .                                   00252
        ZRO(0)          ZRO(0)          .                                   00253
        ZRO(0)          ZRO(0)          .                                   00254
        ZRO(0)          ZRO(0)          .                                   00255
        ZRO(0)          ZRO(0)          .                                   00256
        ZRO(0)          ZRO(0)          .                                   00257
        ZRO(0)          ZRO(0)          .                                   00260
        ZRO(0)          ZRO(0)          .                                   00261
        ZRO(0)          ZRO(0)          .                                   00262
        ZRO(0)          ZRO(0)          .                                   00263
        ZRO(0)          ZRO(0)          .                                   00264
        ZRO(0)          ZRO(0)          .                                   00265
        ZRO(0)          ZRO(0)          .                                   00266
        ZRO(0)          ZRO(0)          .                                   00267
        ZRO(0)          ZRO(0)          .                                   00270
        ZRO(0)          ZRO(0)          .                                   00271
        ZRO(0)          ZRO(0)          .                                   00272
        ZRO(0)          ZRO(0)          .                                   00273
        ZRO(0)          ZRO(0)          .                                   00274
        ZRO(0)          ZRO(0)          .                                   00275
        ZRO(0)          ZRO(0)          .                                   00276
        ZRO(0)          ZRO(0)          .                                   00277

* WRITE BUFFER

        ZRO(0)          ZRO(0)          . WRITE BUFFER                      00300
        ZRO(0)          ZRO(0)          .                                   00301
        ZRO(0)          ZRO(0)          .                                   00302
        ZRO(0)          ZRO(0)          .                                   00303
        ZRO(0)          ZRO(0)          .                                   00304
        ZRO(0)          ZRO(0)          .                                   00305
        ZRO(0)          ZRO(0)          .                                   00306
        ZRO(0)          ZRO(0)          .                                   00307
        ZRO(0)          ZRO(0)          .                                   00310
        ZRO(0)          ZRO(0)          .                                   00311
        ZRO(0)          ZRO(0)          .                                   00312
        ZRO(0)          ZRO(0)          .                                   00313
        ZRO(0)          ZRO(0)          .                                   00314
        ZRO(0)          ZRO(0)          .                                   00315
        ZRO(0)          ZRO(0)          .                                   00316
        ZRO(0)          ZRO(0)          .                                   00317
```

```
ZRO(0)          ZRO(0)          .          00320
ZRO(0)          ZRO(0)          .          00321
ZRO(0)          ZRO(0)          .          00322
ZRO(0)          ZRO(0)          .          00323
ZRO(0)          ZRO(0)          .          00324
ZRO(0)          ZRO(0)          .          00325
ZRO(0)          ZRO(0)          .          00326
ZRO(0)          ZRO(0)          .          00327
ZRO(0)          ZRO(0)          .          00330
ZRO(0)          ZRO(0)          .          00331
ZRO(0)          ZRO(0)          .          00332
ZRO(0)          ZRO(0)          .          00333
ZRO(0)          ZRO(0)          .          00334
ZRO(0)          ZRO(0)          .          00335
ZRO(0)          ZRO(0)          .          00336
ZRO(0)          ZRO(0)          .          00337
ZRO(0)          ZRO(0)          .          00340
ZRO(0)          ZRO(0)          .          00341
ZRO(0)          ZRO(0)          .          00342
ZRO(0)          ZRO(0)          .          00343
ZRO(0)          ZRO(0)          .          00344
ZRO(0)          ZRO(0)          .          00345
ZRO(0)          ZRO(0)          .          00346
ZRO(0)          ZRO(0)          .          00347
ZRO(0)          ZRO(0)          .          00350
ZRO(0)          ZRO(0)          .          00351
ZRO(0)          ZRO(0)          .          00352
ZRO(0)          ZRO(0)          .          00353
ZRO(0)          ZRO(0)          .          00354
ZRO(0)          ZRO(0)          .          00355
ZRO(0)          ZRO(0)          .          00356
ZRO(0)          ZRO(0)          .          00357
ZRO(0)          ZRO(0)          .          00360
ZRO(0)          ZRO(0)          .          00361
ZRO(0)          ZRO(0)          .          00362
ZRO(0)          ZRO(0)          .          00363
ZRO(0)          ZRO(0)          .          00364
ZRO(0)          ZRO(0)          .          00365
ZRO(0)          ZRO(0)          .          00366
ZRO(0)          ZRO(0)          .          00367
ZRO(0)          ZRO(0)          .          00370
ZRO(0)          ZRO(0)          .          00371
ZRO(0)          ZRO(0)          .          00372
ZRO(0)          ZRO(0)          .          00373
ZRO(0)          ZRO(0)          .          00374
ZRO(0)          ZRO(0)          .          00375
ZRO(0)          ZRO(0)          .          00376
ZRO(0)          ZRO(0)          .          00377
ZRO(0)          ZRO(0)          .          00400
```

```
ZRO(0)          ZRO(0)          .                                   00401
ZRO(0)          ZRO(0)          .                                   00402
ZRO(0)          ZRO(0)          .                                   00403
ZRO(0)          ZRO(0)          .                                   00404
ZRO(0)          ZRO(0)          .                                   00405
ZRO(0)          ZRO(0)          .                                   00406
ZRO(0)          ZRO(0)          .                                   00407
ZRO(0)          ZRO(0)          .                                   00410
ZRO(0)          ZRO(0)          .                                   00411
ZRO(0)          ZRO(0)          .                                   00412
ZRO(0)          ZRO(0)          .                                   00413
ZRO(0)          ZRO(0)          .                                   00414
ZRO(0)          ZRO(0)          .                                   00415
ZRO(0)          ZRO(0)          .                                   00416
ZRO(0)          ZRO(0)          .                                   00417
ZRO(0)          ZRO(0)          .                                   00420
ZRO(0)          ZRO(0)          .                                   00421
ZRO(0)          ZRO(0)          .                                   00422
ZRO(0)          ZRO(0)          .                                   00423
ZRO(0)          ZRO(0)          .                                   00424
ZRO(0)          ZRO(0)          .                                   00425
ZRO(0)          ZRO(0)          .                                   00426
ZRO(0)          ZRO(0)          .                                   00427
ZRO(0)          ZRO(0)          .                                   00430
ZRO(0)          ZRO(0)          .                                   00431
ZRO(0)          ZRO(0)          .                                   00432
ZRO(0)          ZRO(0)          .                                   00433
ZRO(0)          ZRO(0)          .                                   00434
ZRO(0)          ZRO(0)          .                                   00435
ZRO(0)          ZRO(0)          .                                   00436
ZRO(0)          ZRO(0)          .                                   00437
ZRO(0)          ZRO(0)          .                                   00440
ZRO(0)          ZRO(0)          .                                   00441
ZRO(0)          ZRO(0)          .                                   00442
ZRO(0)          ZRO(0)          .                                   00443
ZRO(0)          ZRO(0)          .                                   00444
ZRO(0)          ZRO(0)          .                                   00445
ZRO(0)          ZRO(0)          .                                   00446
ZRO(0)          ZRO(0)          .                                   00447
ZRO(0)          ZRO(0)          .                                   00450
ZRO(0)          ZRO(0)          .                                   00451
ZRO(0)          ZRO(0)          .                                   00452
ZRO(0)          ZRO(0)          .                                   00453
ZRO(0)          ZRO(0)          .                                   00454
ZRO(0)          ZRO(0)          .                                   00455
ZRO(0)          ZRO(0)          .                                   00456
ZRO(0)          ZRO(0)          .                                   00457
ZRO(0)          ZRO(0)          .                                   00460
ZRO(0)          ZRO(0)          .                                   00461
```

```
        ZRO(0)          ZRO(0)          .                                   00462
        ZRO(0)          ZRO(0)          .                                   00463
        ZRO(0)          ZRO(0)          .                                   00464
        ZRO(0)          ZRO(0)          .                                   00465
        ZRO(0)          ZRO(0)          .                                   00466
        ZRO(0)          ZRO(0)          .                                   00467
        ZRO(0)          ZRO(0)          .                                   00470
        ZRO(0)          ZRO(0)          .                                   00471
        ZRO(0)          ZRO(0)          .                                   00472
        ZRO(0)          ZRO(0)          .                                   00473
        ZRO(0)          ZRO(0)          .                                   00474
        ZRO(0)          ZRO(0)          .                                   00475
        ZRO(0)          ZRO(0)          .                                   00476
        ZRO(0)          ZRO(0)          .                                   00477
        ZRO(0)          ZRO(0)          .                                   00500
        ZRO(0)          ZRO(0)          .                                   00501
        ZRO(0)          ZRO(0)          .                                   00502
        ZRO(0)          ZRO(0)          .                                   00503
        ZRO(0)          ZRO(0)          .                                   00504
        ZRO(0)          ZRO(0)          .                                   00505
        ZRO(0)          ZRO(0)          .                                   00506
        ZRO(0)          ZRO(0)          .                                   00507
        ZRO(0)          ZRO(0)          .                                   00510
        ZRO(0)          ZRO(0)          .                                   00511
        ZRO(0)          ZRO(0)          .                                   00512
        ZRO(0)          ZRO(0)          .                                   00513
        ZRO(0)          ZRO(0)          .                                   00514
        ZRO(0)          ZRO(0)          .                                   00515
        ZRO(0)          ZRO(0)          .                                   00516
        ZRO(0)          ZRO(0)          .                                   00517
        ZRO(0)          ZRO(0)          .                                   00520
        ZRO(0)          ZRO(0)          .                                   00521
        ZRO(0)          ZRO(0)          .                                   00522
        ZRO(0)          ZRO(0)          .                                   00523
        ZRO(0)          ZRO(0)          .                                   00524
        ZRO(0)          ZRO(0)          .                                   00525
        ZRO(0)          ZRO(0)          .                                   00526
        ZRO(0)          ZRO(0)          .                                   00527
        ZRO(0)          ZRO(0)          .                                   00530
* LISTABLE OUTPUT BUFFER
        ZRO(0)          ZRO(0)          . MONITOR OUTPUT BUFFER             00531
        ZRO(0)          ZRO(0)          .                                   00532
        ZRO(0)          ZRO(0)          .                                   00533
        ZRO(0)          ZRO(0)          .                                   00534
        ZRO(0)          ZRO(0)          .                                   00535
        ZRO(0)          ZRO(0)          .                                   00536
        ZRO(0)          ZRO(0)          .                                   00537
```

```
        ZRO(0)          ZRO(0)          .                               00540
        ZRO(0)          ZRO(0)          .                               00541
        ZRO(0)          ZRO(0)          .                               00542
        ZRO(0)          ZRO(0)          .                               00543
        ZRO(0)          ZRO(0)          .                               00544
        ZRO(0)          ZRO(0)          .                               00545
        ZRO(0)          ZRO(0)          .                               00546
        ZRO(0)          ZRO(0)          .                               00547

* PUNCHED CARD BUFFER

        ZRO(0)          ZRO(0)          . MONITOR PUNCH BUFFER          00550
        ZRO(0)          ZRO(0)          .                               00551
        ZRO(0)          ZRO(0)          .                               00552
        ZRO(0)          ZRO(0)          .                               00553
        ZRO(0)          ZRO(0)          .                               00554
        ZRO(0)          ZRO(0)          .                               00555
        ZRO(0)          ZRO(0)          .                               00556
        ZRO(0)          ZRO(0)          .                               00557
        ZRO(0)          ZRO(0)          .                               00560
        ZRO(0)          ZRO(0)          .                               00561
        ZRO(0)          ZRO(0)          .                               00562
        ZRO(0)          ZRO(0)          .                               00563
        ZRO(0)          ZRO(0)          .                               00564
        ZRO(0)          ZRO(0)          .                               00565
        ZRO(0)          ZRO(0)          .                               00566
        ZRO(0)          ZRO(0)          .                               00567
        ZRO(0)          ZRO(0)          .                               00570
        ZRO(0)          ZRO(0)          .                               00571
        ZRO(0)          ZRO(0)          .                               00572
        ZRO(0)          ZRO(0)          .                               00573
        ZRO(0)          ZRO(0)          .                               00574
        ZRO(0)          ZRO(0)          .                               00575
        ZRO(0)          ZRO(0)          .                               00576
        ZRO(0)          ZRO(0)          .                               00577

* STACKED JOB BUFFER

        ZRO(0)          ZRO(0)          . MONITOR INPUT BUFFER          00600
        ZRO(0)          ZRO(0)          .                               00601
        ZRO(0)          ZRO(0)          .                               00602
        ZRO(0)          ZRO(0)          .                               00603
        ZRO(0)          ZRO(0)          .                               00604
        ZRO(0)          ZRO(0)          .                               00605
        ZRO(0)          ZRO(0)          .                               00606
        ZRO(0)          ZRO(0)          .                               00607
        ZRO(0)          ZRO(0)          .                               00610
```

```
        ZRO(0)          ZRO(0)          .                                   00611
        ZRO(0)          ZRO(0)          .                                   00612
        ZRO(0)          ZRO(0)          .                                   00613
        ZRO(0)          ZRO(0)          .                                   00614
        ZRO(0)          ZRO(0)          .                                   00615
        ZRO(0)          ZRO(0)          .                                   00616
        ZRO(0)          ZRO(0)          .                                   00617
        ZRO(0)          ZRO(0)          .                                   00620
        ZRO(0)          ZRO(0)          .                                   00621
        ZRO(0)          ZRO(0)          .                                   00622
        ZRO(0)          ZRO(0)          .                                   00623
        ZRO(0)          ZRO(0)          .                                   00624
        ZRO(0)          ZRO(0)          .                                   00625
        ZRO(0)          ZRO(0)          .                                   00626
        ZRO(0)          ZRO(0)          .                                   00627

* PROGRAM START TABLE

OZ      ZRO(7)          ZRO(1M)         . MONITOR                           00630
        ZRO(7)          ZRO(1M)         . EEEND                             00631
        ZRO(2)          ZRO(1S)         . INPUT                             00632
        ZRO(2)          ZRO(2S)         . OUTPUT                            00633
        ZRO(0)          ZRO(1T)         . CLOCK                             00634
        ZRO(0)          ZRO(2T)         . STOPCK                            00635
        ZRO(3)          ZRO(1K)         . CONTROL                           00636
        ZRO(0)          ZRO(1SR)        . SR                                00637
        ZRO(9)          ZRO(1C)         . CALL                              00640
        ZRO(6)          ZRO(1J)         . SWITCH                            00641
        ZRO(0)          ZRO(1Q)         . RUN                               00642
        ZRO(0)          ZRO(2Q)         . HOLD                              00643
        ZRO(0)          ZRO(3Q)         . CLEAR                             00644
        ZRO(2)          ZRO(4Q)         . SKIP                              00645
        ZRO(2)          ZRO(3S)         . SET                               00646
        ZRO(0)          ZRO(3T)         . TIME                              00647
        ZRO(2)          ZRO(4T)         . LIMIT                             00650
        ZRO(1)          ZRO(1V)         . REWIND                            00651
        ZRO(1)          ZRO(3V)         . LOCK                              00652
        ZRO(1)          ZRO(5V)         . EFMARK                            00653
        ZRO(1)          ZRO(7V)         . BKSP                              00654
        ZRO(0)          ZRO(00)         . SPARE                             00655
        ZRO(0)          ZRO(0)          . TRANSIENT PROGRAM STARTS          00656
        ZRO(0)          ZRO(0)          .                                   00657
        ZRO(0)          ZRO(0)          .                                   00660
        ZRO(0)          ZRO(0)          .                                   00661
        ZRO(0)          ZRO(0)          .                                   00662
        ZRO(0)          ZRO(0)          .                                   00663
        ZRO(0)          ZRO(0)          .                                   00664
        ZRO(0)          ZRO(0)          .                                   00665
```

```
        ZRO(0)          ZRO(0)          .                                   00666
        ZRO(0)          ZRO(0)          .                                   00667
        ZRO(0)          ZRO(0)          . PAD                               00670
        ZRO(0)          ZRO(G47)        . PRINT MEDIUM DESIGNATOR CELL      00671
0SWT    SEV7(77777B)    SEV7(77777B)    . SATELLITE MONITOR JOB SWITCH      00672
1SWT    SEV7(77777B)    SEV7(77777B)    . 160 LIBRARY CALL LOCK-OUT         00673
1ERSW   ZRO(0)          ZRO(0)          .                                   00674
1TRIG   SLJ(N)          STA(0SWT)       . TRIGGER ROUTINE                   00675
        SLJ(L-1)        ZRO(0)          .     FOR SATELLITE MONITOR         00676
1CALL   SLJ(N)          SLJ(160CL)      . 160 SAT LIBRARY CALL ENTRY        00677
1PCON   EXF(62500B)     SLJ1(L+6)       . RELEASE DIRECT / TEST NO DD65     00700
        EXF(52502B)     EXF(62502B)     . TAKE READ/WRITE CONTROL           00701
        EXF(77010B)     ISK1(1000B)     . SEL. DD65                         00702
        EXF7(77010B)    SLJ(1?)         . SENSE DD65 AVAILABLE              00703
        ENI(0)          OUT(Z+40B)      . CLEAR DD65 CORE                   00704
        ISK1(1000B)     SLJ(L-1)        . LOOP UNTIL CLEAR                  00705
        EXF(62560B)     SLJ(1?)         . GO TO PROGRAM CONTROL             00706

* TIME ADVANCE ROUTINE

1A      EXF(70B)        STA(T+8)        . CLEAR FAULT                       00707
        SSK(Z)          SLJ(Z+7)        . SENSE ONE-SECOND OVERFLOW         00710
        LDA(R12)        STA(Z)          . RESET ONE-SECOND COUNT            00711
        RAO(Z+50B)      SUB(Z+57B)      . INCREMENT TIME AT 50              00712
        AJP2(L+2)       LDA(T+8)        . SENSE LIMIT AT 57                 00713
        SLJ(Z+7)        ZRO(0)          . EXIT THROUGH 7                    00714
2A      EXF(2000B)      ENA(L+3)        . STOP CLOCK                        00715
        SAU(Z+7)        LDA(R11)        . SET TO ISSUE LIMIT INDICATION     00716
        STA(Z+57B)      SLJ(Z+7)        . EXIT THROUGH 7                    00717
        ENA(1)          SAL(2V)         . REWIND LIBRARY TAPE               00720
        SLJ4(1V)        ZRO(0)          .                                   00721
        ENA(E16)        ENI(0)          . SET TO INDICATE LIMIT             00722
        SSK(2M)         SLJ(2E)         . SENSE NO MONITOR CONTROL          00723
        SLJ4(6G)        ZRO(0)          . TYPE LIMIT INDICATION             00724
        EXF(1000B)      SLJ(9M)         . RETURN TO MONITOR                 00725

* BCD READ SWITCH

3A      SSK(3M)         SLJ(L+6)        . SENSE FULL MONITOR CONTROL        00726
        SSK(2M)         SLJ(L+3)        . SENSE PARTIAL MONITOR CONTROL     00727
        SLJ4(1R)        ZRO(0)          . GO READ BCD RECORD                00730
        SLJ(3AEX)       ZRO(0)          .                                   00731
        EQS(63B)        SLJ(L-2)        . SENSE CARD REQUEST                00732
        LDA(2M)         SLJ(L-3)        . SET TO READ CARD IMAGE            00733
        LDA(2M)         ENI2(0)         .                                   00734
        THS(9)          SLJ(L+19)       . SENSE STACKED-JOB TAPE            00735
        THS(5)          SLJ(L+2)        . SENSE FIRST OR SECOND BANK        00736
```

```
        INA(3200B)      SLJ(L+2)        .                                       00737
        INA(5174B)      ENI2(1)         .                                       00740
        ALS(3)          INA(2)          .                                       00741
        SAU(L+9)        SAU(L+11)       . SET TO SELECT TAPE UNIT               00742
        LDQ(R6)         EXF7(31B)       . SENSE CHANNELS INACTIVE               00743
        ENI1(9)         EXF7(51B)       .                                       00744
        LDA1(M+39)      STA1(R)         . TRANSFER FROM BUFFER                  00745
        IJP1(L-1)       ENI1(4)         .                                       00746
        STQ1(R+10)      IJP1(L)         .                                       00747
        EXF(62540B)     SLJ4(3G)        . DISASSEMBLE READ BUFFER               00750
        ENQ(M+49)       STQ(Z+3)        .                                       00751
        STQ(Z+5)        IJP2(L+3)       .                                       00752
        EXF(N)          EXF7(32000B)    . READ NEXT BLOCK TO BUFFER             00753
        EXF3(M+39)      SLJ(3AEX)       .                                       00754
        EXF(N)          EXF7(52000B)    .                                       00755
        EXF5(M+39)      EXF7(52000B)    .                                       00756
        ENI(0)          SLJ(3AEX)       .                                       00757
        EQS(63B)        SLJ(L-24)       . SENSE STACKED-JOB DECK                00760
        ENI1(23)        EXF7(11B)       . SENSE CHANNEL INACTIVE                00761
        LDA1(M+39)      STA1(R+80)      . TRANSFER FROM BUFFER                  00762
        IJP1(L-1)       SLJ4(3R)        . GO PROCESS CARD IMAGE                 00763
        EXF(62540B)     ENI1(79)        .                                       00764
        LDA1(R)         INA(-14B)       .                                       00765
        AJP1(L+2)       ENA(40B)        . SENSE ALTERNATE DASH                  00766
        STA1(R)         ENI(0)          .                                       00767
        IJP1(L-3)       ENI(0)          .                                       00770
        ENQ(M+63)       STQ(Z+1)        .                                       00771
        ENA(14001B)     ADD(T+11)       .                                       00772
        SAU(L+1)        EXF7(24004B)    .                                       00773
        EXF(N)          EXF1(M+39)      . READ NEXT CARD TO BUFFER              00774
        ENI(0)          EXF7(24004B)    .                                       00775
3AEX    EXF(62560B)     ENA(+0)         . RELEASE COMM FLAG 1                   00776
        STA(1ERSW)      SLJ(Z+11B)      . CLEAR ERROR FLAG / EXIT VIA 7         00777

* BCD WRITE SWITCH

4A      SSK(3M)         SLJ(L+11)       . SENSE FULL MONITOR CONTROL            01000
        SSK(2M)         SLJ(L+7)        . SENSE PARTIAL MONITOR CONTROL         01001
        EQS(63B)        SLJ(L+4)        . SENSE CARD REQUEST                    01002
        LDQ(G63)        LDL(77B)        .                                       01003
        INA(-12B)       AJP(L+12)       . SENSE CARD-PUNCH REASSIGNMENT         01004
        ENA(63B)        ENI(0)          .                                       01005
        SLJ4(1W)        ZRO(0)          . GO WRITE BCD RECORD                   01006
        SLJ(4AEX)       ZRO(0)          .                                       01007
        EQS(2M)         SLJ(L-2)        . SENSE STACKED-JOB MEDIUM              01010
        THS(9)          SLJ(L-3)        .                                       01011
        ENA(27B)        SLJ(L-4)        .                                       01012
        EQS(63B)        SLJ(L+13)       . SENSE CARD REQUEST                    01013
```

```
        LDA(6M)          AJP(L+12)        . SENSE PUNCH-CARD MEDIUM          01014
        INA(-63B)        AJP(L+3)         . SENSE ON-LINE PUNCHING           01015
        ENA(0)           SLJ4(8G)         . SENSE ALL CHANNELS INACTIVE      01016
        LDA(6M)          SLJ(L-9)         .                                  01017
        SLJ4(3W)         ZRO(0)           . GO FORM CARD IMAGE               01020
        ENI1(23)         EXF7(21B)        . SENSE CHANNEL INACTIVE           01021
        LDA1(W)          STA1(M+15)       . TRANSFER TO BUFFER               01022
        IJP1(L-1)        SLJ4(1FLAG)      .                                  01023
        ENA(M+39)        EXF7(14004B)     .                                  01024
        EXF(24004B)      SAL(Z+2)         .                                  01025
        EXF2(M+15)       EXF7(14004B)     . PUNCH NEXT CARD FROM BUFFER      01026
        ENI(0)           SLJ(4AEX)        .                                  01027
        LDA(3M)          ENI2(0)          .                                  01030
        THS(9)           SLJ(L-19)        . SENSE LISTABLE-OUTPUT TAPE       01031
        THS(5)           SLJ(L+2)         . SENSE FIRST OR SECOND BANK       01032
        INA(4200B)       SLJ(L+2)         .                                  01033
        INA(6174B)       ENI2(1)          .                                  01034
        ALS(3)           INA(2)           .                                  01035
        SAU(L+7)         SAU(L+9)         . SET TO SELECT TAPE UNIT          01036
        EXF(62540B)      SLJ4(4G)         . ASSEMBLE WRITE BUFFER            01037
        ENI1(14)         EXF7(41B)        . SENSE CHANNELS INACTIVE          01040
        ENQ(M+15)        EXF7(51B)        .                                  01041
        LDA1(W)          STA1(M)          . TRANSFER TO BUFFER               01042
        IJP1(L-1)        STQ(Z+4)         .                                  01043
        STQ(Z+6)         IJP2(_+3)        .                                  01044
        EXF(N)           EXF7(+2000B)     . WRITE NEXT BLOCK FROM BUFFER     01045
        EXF4(M)          SLJ(4AEX)        .                                  01046
        EXF(N)           EXF7(62000B)     .                                  01047
        EXF6(M)          EXF7(62000B)     .                                  01050
4AEX    EXF(62560B)      ENI(0)           .                                  01051
        SSK(OSWT)        SLJ(L+2)         . ERRORS TO SATELLITE ROUTINE      01052
        SLJ(Z+12B)       ZRO(0)           .                                  01053
        SSK(1ERSW)       SLJ(Z+12B)       .                                  01054
        LDA(3M)          INA(-56B)        .                                  01055
        AJP(Z+12B)       ENA(Z+12B)       .                                  01056
        SAU(1W)          SLJ(7W)          .                                  01057

* BIN READ SWITCH

5A      SSK(2M)          SLJ4(12A)        . SENSE NO MONITOR CONTROL         01060
        SLJ4(11)         ZRO(0)           . GO READ BINARY RECORD            01061
        SLJ(Z+13B)       ZRO(0)           . EXIT THROUGH 13                  01062

* BIN WRITE SWITCH

6A      SSK(2M)          SLJ4(3G)         . SENSE NO MONITOR CONTROL         01063
        SLJ4(10)         ZRO(0)           . GO WRITE BINARY RECORD           01064
```

```
      SLJ(Z+14B)        ZRO(0)          . EXIT THROUGH 14                    01065

* BACKSPACE SWITCH

7A    SSK(2M)           SLJ4(12A)       . SENSE NO MONITOR CONTROL           01066
      SAL(8V)           SLJ4(7V)        . BACKSPACE                          01067
      SLJ(Z+15B)        ZRO(0)          . EXIT THROUGH 15                    01070

* REWIND SWITCH

8A    SSK(2M)           SLJ4(8G)        . SENSE NO MONITOR CONTROL           01071
      SAL(2V)           SLJ4(1V)        . REWIND                             01072
      SLJ(Z+16B)        ZRO(0)          . EXIT THROUGH 16                    01073

* ENDFILE SWITCH

9A    SSK(2M)           SLJ4(12A)       . SENSE NO MONITOR CONTROL           01074
      SAL(6V)           SLJ4(5V)        . ENDFILE                            01075
      SLJ(Z+17B)        ZRO(0)          . EXIT THROUGH 17                    01076

* CHAIN CONNECTOR

10A   SAL(L+1)          INA(1)          . SET TO PICK ARGUMENTS              01077
      SAL(L+1)          LDA7(N)         . TRANSFER ARGUMENTS                 01100
      STA(3C)           LDA7(N)         .                                    01101
      STA(2C)           SLJ4(1C)        . CALL NEXT LINK                     01102
      SLJ(1Q)           ZRO(0)          . GO EXECUTE ASSEMBLED PROGRAM       01103

* BCD MESSAGE SWITCH

11A   SSK(2M)           SLJ(L+3)        . SENSE NO MONITOR CONTROL           01104
      SLJ4(6G)          ZRO(0)          . GO OUTPUT BCD MESSAGE              01105
      SLJ(Z+25B)        ZRO(0)          . EXIT THROUGH 25                    01106
      SAU(L+1)          SAL(L+2)        .                                    01107
      LDA(N)            SCM(R3)         .                                    01110
      AJP(Z+25B)        ENA(N)          . SENSE FORTRAN-E                    01111
      SLJ(2E)           ZRO(0)          . EXIT TO STACK MESSAGE              01112

* TAPE ASSIGNMENT CHECK ROUTINE

12A   SLJ(N)           ENI1(3)         .                                    01113
      EQS(1)            SLJ(L+2)        . SENSE LIBRARY TAPE                 01114
      SLJ(12A)          ZRO(0)          .                                    01115
```

```
        ALS(24)          LQC(77777B)      .                                  01116
        MEQ1(C+8)        SLJ(L+3)         .                                  01117
        ALS(24)          SLJ4(8G)         .                                  01120
        SLJ(L-6)         ZRO(0)           .                                  01121
        LIU1(L-7)        INI1(1B)         .                                  01122
        SIU1(L-8)        SLJ(L-8)         .                                  01123

* CALL ROUTINE

1C      SLJ(N)           SLJ(L+10)        . EXIT/ENTRY                       01124
2C      ZRO(0)           ZRO(0)           . LIBRARY MEDIUM                   01125
3C      ZRO(0)           ZRO(0)           . PROGRAM-NAME LIST                01126
        ZRO(0)           ZRO(0)           .                                  01127
        ZRO(0)           ZRO(0)           .                                  01130
        ZRO(0)           ZRO(0)           .                                  01131
        ZRO(0)           ZRO(0)           .                                  01132
        ZRO(0)           ZRO(0)           .                                  01133
        ZRO(0)           ZRO(0)           .                                  01134
        ZRO(0)           ZRO(0)           .                                  01135
        LDQ(4M)          QJP1(L+3)        . SENSE MONITOR CONTROL            01136
        LDA(2C)          SAL(2V)          . REWIND LIBRARY MEDIUM            01137
        SLJ4(1V)         ZRO(0)           .                                  01140
        LDA(C+4)         STA(C+5)         . CLEAR TRANSIENT PROGRAMS         01141
        LDA(C+2)         STA(C+3)         . RESET PROGRAM BIAS               01142
        ENI1(7)          ENI5(1)          .                                  01143
        LDA1(3C)         ENQ(0)           . POSITION NAMES                   01144
        LRS(6)           AJP1(_)          .                                  01145
        STQ1(3C)         IJP1(L-2)        . LOOP                             01146
4C      LDA(2C)          SLJ4(1I)         . READ BINARY BLOCK                01147
5C      LDA(R)           AJP(L-1)         .                                  01150
        LDQ(3C)          ENI(0)           .                                  01151
        IJP5(L+1)        SLJ(L+2)         .                                  01152
        QJP1(L+1)        STA(3C)          . SENSE NO PROGRAM-NAME LIST       01153
        SCM(R10)         AJP(9C)          . SENSE END OF LIBRARY             01154
        LDA(R)           ENI1(3)          .                                  01155
        EQS1(3C)         SLJ(4C)          . LOOP TO CALLED NAME              01156
        LIL3(C+5)        STA3(PO)         . STORE PROGRAM NAME               01157
        LDQ(R+1)         LDL(77777B)      . PROCESS PARAMETERS               01160
        STA(T)           QRS(24)          .                                  01161
        LDL(77777B)      ADD(T)           .                                  01162
        ADD(C+3)         SUB(77777B)      .                                  01163
        AJP2(8C)         LDQ(R+1)         . SENSE OVERFLOW OF MEMORY         01164
        QLS(9)           ENI(0)           .                                  01165
        LDL(777B)        ALS(24)          .                                  01166
        ADD(C+3)         STA3(0Z)         .                                  01167
        QLS(15)          LDL(77777B)      .                                  01170
        ADD(C+3)         STA(T+7)         .                                  01171
        LDA(C+3)         STA(T+3)         .                                  01172
```

```
        ENA(0)          STA1(3C)        . CLEAR NAME FROM LIST          01173
        RAO(C+5)        ENI4(3)         .                               01174
6C      ENI1(0)         ENI(0)          .                               01175
        LDA1(R+6)       STA7(T+3)       . COPY WORD                     01176
        SSH(R+2)        SLJ(L+3)        . SENSE UPPER RELATIVE ADDRESS  01177
        LDA(C+3)        ALS(24)         .                               01200
        RAD7(T+3)       ENI(0)          . MODIFY UPPER ADDRESS          01201
        SSH(R+4)        SLJ(L+2)        . SENSE LOWER RELATIVE ADDRESS  01202
        LDA(C+3)        RAD7(T+3)       . MODIFY LOWER ADDRESS          01203
        RAO(T+3)        ENI(0)          . ADVANCE PROGRAM ADDRESS       01204
        ISK1(47)        SLJ(L-7)        . LOOP                          01205
        LDA(2C)         SLJ4(11)        . READ BINARY BLOCK             01206
        LDA(R)          AJP(6C)         . SENSE END OF PROGRAM          01207
        LDA(T+7)        STA(C+3)        . ADVANCE BIAS                  01210
        LDA4(3C)        AJP1(5C)        . SENSE COMPLETION OF CALL      01211
        ISK4(7)         SLJ(L-1)        .                               01212
7C      LDA(2C)         SLJ4(9CA)       . REWIND LIBRARY MEDIUM         01213
        SLJ4(1V)        ZRO(0)          .                               01214
        SLJ4(10C)       ZRO(0)          .                               01215
        SLJ(1C)         ZRO(0)          . EXIT                          01216
8C      LDA(2C)         SLJ4(9CA)       . REWIND LIBRARY MEDIUM         01217
        SLJ4(1V)        ZRO(0)          .                               01220
        SLJ4(10C)       ZRO(0)          .                               01221
        ENA(E3)         SLJ(2E)         . ERROR EXIT                    01222
9C      ENI1(7)         ENI(0)          .                               01223
        LDA1(3C)        AJP1(L+2)       .                               01224
        IJP1(L-1)       SLJ(7C)         .                               01225
        LDA1(3C)        STA(E7)         .                               01226
        ENA(0)          STA1(3C)        .                               01227
        LDA(2C)         SLJ4(9CA)       . REWIND LIBRARY MEDIUM         01230
        SLJ4(1V)        ZRO(0)          .                               01231
        SLJ4(10C)       ZRO(0)          .                               01232
        ENA(E5)         SLJ(2E)         . ERROR EXIT                    01233
9CA     SLJ(N)          SAL(2V)         . RESET 160 LIB CALL LOCK-OUT   01234
        ENA(-0)         STA(1SWT)       .                               01235
        EXF(53511B)     SLJ(L-2)        .                               01236

* INITIALIZE FOR NEXT CALL

10C     SLJ(N)          ENA(S)          . EXIT/ENTRY                    01237
        STA(C+2)        ENA(J)          .                               01240
        STA(C+4)        ENI1(7)         .                               01241
        ENA(0)          STA1(3C)        .                               01242
        IJP1(L-1)       STA(Z+47B)      .                               01243
        SLJ(10C)        ZRO(0)          . EXIT                          01244

* ERROR MESSAGE HANDLER
```

```
1E      SAU(L+1)          SLJ4(5G)        . INDICATE SYSTEM FAILURE          01245
        ENA(N)            LDQ(3M)         .                                  01246
        QJP2(2E)          SLJ(10M)        . SENSE FULL MONITOR CONTROL       01247
2E      AJP(10M)          LDQ(4K)         . SENSE ERROR EXIT FROM COMPILER   01250
        QJP4(6G)          SLJ4(3E)        . GO OUTPUT ERROR MESSAGE          01251
        EXF(1000B)        SLJ(10M)        . EXIT TO MONITOR                  01252

* OUTPUT PACKED BCD MESSAGE

3E      SLJ(N)            SAU(L+2)        . EXIT/ENTRY                       01253
        ENA(0)            SLJ4(3G)        . GO SENSE CHANNELS INACTIVE       01254
        ENI1(N)           ENI2(0)         . STRING OUT BCD CODE              01255
        ENA(20B)          STA(W)          .                                  01256
        ENA(0)            ENI3(0)         .                                  01257
        LDQ1(Z)           LLS(6)          .                                  01260
        AJP(L+4)          STA2(W+1)       .                                  01261
        ENA(0)            INI2(1)         .                                  01262
        ISK3(7)           SLJ(L+6)        .                                  01263
        INI1(1)           SLJ(L-4)        .                                  01264
        ENA(20B)          STA2(W+1)       .                                  01265
        ISK2(119)         SLJ(L-1)        .                                  01266
        LDA(4K)           SLJ4(1W)        . OUTPUT MESSAGE                   01267
        EXF(1000B)        SLJ(3E)         . EXIT                             01270
        LLS(6)            SLJ(L-8)        .                                  01271

* INPUT INDEX REGISTER LOADER

1G      SLJ(N)            STA(T)          . EXIT/ENTRY                       01272
        LIL1(T)           LDQ1(30)        . TABLE ITEMS TO INDEX REGISTERS   01273
        QRS(6)            LDL(73)         .                                  01274
        SAL(L+2)          QRS(3)          .                                  01275
        LDL(7B)           SAU(L+1)        .                                  01276
        ENI1(N)           ENI2(N)         .                                  01277
        SLJ(L-6)          ZRO(0)          . EXIT                             01300

* OUTPUT INDEX REGISTER LOADER

2G      SLJ(N)            STA(T)          . EXIT/ENTRY                       01301
        LIL1(T)           LDQ1(30)        . TABLE ITEMS TO INDEX REGISTERS   01302
        LDL(7B)           SAL(L+2)        .                                  01303
        QRS(3)            LDL(73)         .                                  01304
        SAU(L+1)          ENI2(N)         .                                  01305
        ENI1(N)           SLJ(L-5)        . EXIT                             01306
```

```
* DISASSEMBLE READ BUFFER

3G      SLJ(N)              SIL2(L+10)      . EXIT/ENTRY                        01307
        ENA(R+112)          SAL(L+6)        . DISASSEMBLE CHARACTERS            01310
        ENI2(14)            ENI(0)          .                                   01311
        LDQ2(R)             ENI1(0)         .                                   01312
        ENA(0)              LLS(6)          .                                   01313
        EQS(14B)            SLJ(L+2)        . SENSE ALTERNATE DASH              01314
        ENA(40B)            ENI(0)          .                                   01315
        ENI(0)              STA1(N)         . STORE CHARACTER                   01316
        ISK1(7)             SLJ(L-4)        . LOOP                              01317
        ENA(8)              RSB(L-2)        .                                   01320
        IJP2(L-7)           ENI2(N)         .                                   01321
        ENA(0)              STA(R+120)      .                                   01322
        SLJ(3G)             ZRO(0)          . EXIT                              01323

* ASSEMBLE WRITE BUFFER

4G      SLJ(N)              ENI1(0)         . EXIT/ENTRY                        01324
        ENA(W)              SAL(L+8)        . ASSEMBLE CHARACTERS               01325
        LDA1(W)             ALS(6)          .                                   01326
        ADD1(W+1)           ALS(6)          .                                   01327
        ADD1(W+2)           ALS(6)          .                                   01330
        ADD1(W+3)           ALS(6)          .                                   01331
        ADD1(W+4)           ALS(6)          .                                   01332
        ADD1(W+5)           ALS(6)          .                                   01333
        ADD1(W+6)           ALS(6)          .                                   01334
        ADD1(W+7)           STA(N)          . STORE WORD                        01335
        INI1(7)             RAO(L-1)        .                                   01336
        ISK1(119)           SLJ(L-9)        . LOOP                              01337
        SLJ(4G)             ZRO(0)          . EXIT                              01340

* SENSE TYPEWRITER FOR ACTION

5G      SLJ(N)              SLJ4(7G)        . EXIT/ENTRY                        01341
        ENA(T+1)            SLJ4(1FLAG)     .                                   01342
        SAL(Z+1)            EXF7(11B)       .                                   01343
        EXF(11100B)         ENA(1)          .                                   01344
        EXF1(T)             ISK1(100B)      .                                   01345
        EXF7(111008)        SLJ(L+2)        . SENSE CARRIAGE RETURN             01346
        EXF1(T+1)           SLJ(5GEX)       .                                   01347
        EXF7(11B)           SLJ(L-2)        . SENSE CHARACTER                   01350
        LDA(T)              EXF(10000B)     .                                   01351
        INA(-4)             AJP(5GEX)       . SENSE SPACE                       01352
        INA(-36B)           AJP(1JM)        . SENSE PERIOD                      01353
        INA(13B)            AJP1(L-8)       . SENSE X                           01354
        ENA(-1)             ENI(0)          .                                   01355
```

```
5GEX   EXF(62560B)       SLJ(5G)          .                              01356

5GG    SLJ4(5G)          ZRO(0)           . GO SENSE TYPEWRITER          01357
       SLJ(Z+31B)        ZRO(0)           . RETURN VIA CELL 31           01360

* BCD TYPEWRITER OUTPUT

6G     SLJ(N)            SAL(L+3)         . EXIT/ENTRY                   01361
       ENA(R5)           SLJ4(7G)         . GET CARRIAGE RETURN          01362
       EXF(62540B)       EXF7(21B)        .                              01363
       EXF(21110B)       LDA(N)           . SELECT CHARACTER MODE        01364
       ENI(0)            STA(T+1)         . PICK NEXT WORD               01365
       ENI2(7)           ENI(0)           .                              01366
       LDQ(T+1)          QLS(6)           .                              01367
       STQ(T+1)          LDL(77B)         .                              01370
       ENQ(0)            AJP(6GEX)        . SENSE END OF CODE            01371
       STA(T)            LIL1(T)          .                              01372
       LDQ1(K0)          QLS(6)           . TABLE LOOKUP                 01373
       LDL(77B)          AJP1(L+4)        .                              01374
       EXF7(11141B)      SLJ(L+6)         . SENSE CARRIAGE DOWN          01375
       ENA(57B)          STA(T)           .                              01376
       SLJ(L+3)          ZRO(0)           .                              01377
       EXF7(11140B)      SLJ(L+3)         . SENSE CARRIAGE UP            01400
       ENA(47B)          STA(T)           .                              01401
       EXF2(T)           EXF7(21B)        . SHIFT CARRIAGE               01402
       QLS(6)            STQ(T)           .                              01403
       EXF2(T)           EXF7(21B)        . PRINT CHARACTER              01404
       ENI(0)            SLJ4(1DLAY)      . GO SLOW DOWN TYPEWRITER      01405
       IJP2(L-15)        RAO(L-18)        . SENSE END OF WORD            01406
       SLJ(L-19)         ZRO(0)           . LOOP                         01407
6GEX   EXF(62560B)       SLJ(6G)          .                              01410

* COMM FLAG1 SET AND DELAY ROUTINE

1FLAG  SLJ(N)            EXF(62540B)      . SET COMM FLAG1               01411
       SIU1(L+4)         SLJ4(1DLAY)      . GO DELAY                     01412
       SLJ(L-2)          ZRO(0)           . RETURN                       01413

* TYPEWRITER SLOW-DOWN LOOP

1DLAY  SLJ(N)            SIU1(_+2)        . EXIT/ENTRY                   01414
       ENI1(0)           ISK1(100B)       . DELAY                        01415
       ENI1(N)           SLJ(L-2)         . RE-STORE INDEX / GO EXIT     01416

* TYPEWRITER PRINT ROUTINE
```

```
7G      SLJ(N)              EXF7(21B)          . EXIT/ENTRY                                01417
        ENI(0)              SLJ4(1FLAG)        . GO SET COMM FLAG1                         01420
        SAL(L+5)            EXF(21110B)        .                                           01421
        ENA(T+1)            SAL(Z+2)           .                                           01422
        EXF7(11141B)        SLJ(L+3)           . SENSE CARRIAGE UP                         01423
        ENA(57B)            STA(T)             .                                           01424
        EXF2(T)             EXF7(21B)          . SHIFT CARRIAGE DOWN                       01425
        ENI1(7)             LDQ(N)             . PICK NEXT WORD                            01426
        QLS(6)              LDL(77B)           .                                           01427
        AJP(7GEX)           STA(T)             . SENSE END OF PRINT CODE                   01430
        EXF2(T)             EXF7(21B)          . PRINT CHARACTER                           01431
        ENI(0)              SLJ4(1DLAY)        . GO TO SLOW DOWN TYPEWRITE                 01432
        IJP1(L-4)           RAO(L-5)           . SENSE EIGHT CHARACTERS                    01433
        SLJ(L-6)            ZRO(0)             . LOOP                                      01434
7GEX    EXF(62560B)         SLJ(7G)            .                                           01435

* ASSIGN TAPE MEDIUM AND/OR SENSE ALL CHANNELS INACTIVE

8G      SLJ(N)              AJP(L+15)          . EXIT/ENTRY                                01436
        EQS(1)              SLJ(L+2)           . SENSE LIBRARY TAPE                        01437
        SLJ(8G)             ZRO(0)             .                                           01440
        SSK(3M)             SLJ(L+3)           . SENSE FULL MONITOR CONTROL                01441
        EQS(2M)             SLJ(L-2)           . SENSE STACKED-JOB MEDIUM                  01442
        ENA(27B)            SLJ(8G)            .                                           01443
        LQC(77777B)         STQ(T+1)           .                                           01444
        ALS(24)             ENI1(8)            .                                           01445
        MEQ1(C+8)           SLJ(L+2)           . SENSE PREVIOUS ASSIGNMENT                 01446
        LDA1(C+8)           SLJ(L+6)           . NUMBER TO ACCUMULATOR                     01447
        STA(T)              ENI(0)             .                                           01450
        LDL1(C+8)           AJP(L+3)           . SEARCH FOR FREE TAPE                      01451
        ISK1(7)             SLJ(L-1)           .                                           01452
        ENA(27B)            SLJ(8G)            .                                           01453
        LDA(T)              RAD1(C+8)          . NUMBER TO TABLE AND ACCUMULATOR           01454
        EXF7(11B)           SLJ(L)             . SENSE ALL CHANNELS INACTIVE               01455
        EXF7(21B)           SLJ(L)             .                                           01456
        EXF7(31B)           SLJ(L)             .                                           01457
        EXF7(41B)           SLJ(L)             .                                           01460
        EXF7(51B)           SLJ(L)             .                                           01461
        EXF7(61B)           SLJ(L)             .                                           01462
        SCL(T+1)            SLJ(8G)            . EXIT                                      01463

* READ BINARY RECORD

1I      SLJ(N)              STA(1SWT)          . EXIT/ENTRY                                01464
        EXF(63511B)         INA(-2020B)        .                                           01465
        AJP1(L+1)           INA(63B)           . SENSE CARD REFERENCE                      01466
```

```
        INA(2020B)      SLJ4(1G)        . GET NUMERIC DESIGNATOR                01467
        IJP1(3I)        IJP2(2I)        . SENSE NO MEDIUM REFERENCE             01470
        ENA(0)          ENI1(53)        . ENDFILE BLOCK TO BUFFER               01471
        STA1(R)         IJP1(_)         .                                       01472
        LDA(R10)        STA(R)          .                                       01473
        SLJ(1I)         ZRO(0)          . EXIT                                  01474
2I      ENA2(-5)        AJP1(5I)        . SENSE MEMORY REFERENCE                01475
        LDA(5S)         INA(-1)         . READ MEMORY RECORD                    01476
        SAU(L+2)        ENI(0)          .                                       01477
        LDQ7(6S)        STQ1(R)         .                                       01500
        ISK1(N)         SLJ(L-1)        .                                       01501
        INA(1)          RAD(6S)         .                                       01502
        SLJ(1I)         ZRO(0)          . EXIT                                  01503
3I      IJP1(L+1)       SLJ(4I)         . SENSE TAPE REFERENCE                  01504
        ENA(10X)        SAL(9X)         . READ TAPE RECORD                      01505
        LDA(U1)         ENI(0)          .                                       01506
        IJP1(L+1)       LDA(UJ)         .                                       01507
        IJP1(5I)        ARS(24)         .                                       01510
        STA(T)          ENA(R+54)       .                                       01511
        SAU(2X)         SAU(6X)         .                                       01512
        ENA2(0)         ALS(3)          .                                       01513
        ADD(T)          INA(1)          .                                       01514
        SLJ4(1X)        ZRO(0)          .                                       01515
        SLJ(1I)         ZRO(0)          . EXIT                                  01516
4I      ENA(14002B)     SLJ4(1FLAG)     . SENSE CARD REFERENCE                  01517
        IJP2(L+1)       INA(-1)         .                                       01520
        IJP2(5I)        SAU(L+3)        .                                       01521
        ENA(R)          ENI1(2)         . READ BINARY CARD GROUP                01522
        SAU(L+2)        INA(24)         .                                       01523
        EXF(N)          SAL(Z+1)        .                                       01524
        EXF1(N)         EXF7(24004B)    . READ CARD                             01525
        IJP1(L-3)       ENI(0)          . LOOP                                  01526
        ENA(R+54)       SAL(L+1)        . ASSEMBLE EXTRA BITS                   01527
        ENI2(4)         LDQ(N)          .                                       01530
        LDA1(R)         QLS(8)          .                                       01531
        ADL(R7)         ENI(0)          . SENSE FOR MINUS ZERO                  01532
        AJP1(L+2)       LDL(R7)         .                                       01533
        AJP(L+1)        LDA(-0)         .                                       01534
        STA1(R)         ENI(0)          .                                       01535
        INI1(1)         IJP2(L-5)       .                                       01536
        RAO(L-7)        ENA1(-55)       .                                       01537
        AJP1(L-8)       ENI2(53)        . LOOP                                  01540
        ADD2(R)         IJP2(L)         . GET CHECK SUM                         01541
        EXF(70B)        STA(T)          .                                       01542
        ENI1(70)        LDA1(R)         .                                       01543
        LDQ1(R+1)       QLS(8)          .                                       01544
        LLS(40)         SCM(T)          .                                       01545
        AJP(4IEX)       ENI(0)          . CHECK FOR CORRECT READ                01546
        ENA(T0)         SLJ4(5G)        . TROUBLE EXIT                          01547
```

```
        AJP(4IEX)       SLJ(L-22)       . SENSE FOR ACTION TO TAKE          01550
4IEX    EXF(62560B)     SLJ(1I)         .                                   01551
5I      ENA(E9)         SLJ(1E)         . ERROR EXIT                        01552

* SWITCH SELECTION ROUTINE

1J      SLJ(N)          SLJ(L+7)        . EXIT/ENTRY                        01553
        ZRO(0)          ZRO(0)          . ARGUMENTS                         01554
        ZRO(0)          ZRO(0)          .                                   01555
        ZRO(0)          ZRO(0)          .                                   01556
        ZRO(0)          ZRO(0)          .                                   01557
        ZRO(0)          ZRO(0)          .                                   01560
        ZRO(0)          ZRO(0)          .                                   01561
        ENA(0)          ENI1(5)         .                                   01562
        STA1(Z+51B)     IJP1(L)         . CLEAR SIMULATION LOCATIONS        01563
        LDA1(L-8)       ENI2(6)         .                                   01564
        EQS2(F1)        SLJ(L+3)        . SENSE SWITCH SELECTION            01565
        ENA(-0)         STA2(Z+51B)     .                                   01566
        INI1(1)         SLJ(L-3)        .                                   01567
        ENA(0)          ENI1(5)         .                                   01570
        STA1(L-13)      IJP1(_)         . CLEAR ARGUMENTS                   01571
        SLJ(L-15)       ZRO(0)          . EXIT                              01572

* CONTROL INITIALIZER

1K      SLJ(N)          SLJ(L)          . EXIT/ENTRY                        01573
2K      ZRO(0)          ZRO(0)          . CONTROL MEDIUM                    01574
3K      ZRO(0)          ZRO(0)          . CONTROL STATEMENT OUTPUT MEDIUM   01575
4K      ZRO(0)          ZRO(0)          . ERROR OUTPUT MEDIUM               01576

* MONITOR ROUTINE

1M      ENI(0)          SLJ(L+8)        . ENTRY                             01577
2M      ZRO(0)          ZRO(0)          . STACKED-JOB MEDIUM                01600
3M      ZRO(0)          ZRO(0)          . LISTABLE-OUTPUT MEDIUM            01601
4M      ZRO(0)          ZRO(0)          . COMPILE-MODE INDICATOR            01602
5M      ZRO(0)          ZRO(0)          . JOB-TIME LIMIT                    01603
6M      ZRO(0)          ZRO(0)          . PUNCH-CARD MEDIUM                 01604
7M      ZRO(0)          ZRO(0)          . SOURCE-LANGUAGE INDICATOR         01605
8M      ZRO(0)          ZRO(0)          . INTERMEDIATE-LANGUAGE INDICATOR   01606
        LDA(Z+47B)      AJP(L+4)        . SENSE RECOVERY INDICATOR          01607
        LDA(2M)         SAL(8V)         . BACKSPACE STACKED-JOB MEDIUM      01610
        SLJ4(7V)        ZRO(0)          .                                   01611
        SLJ(L+5)        ZRO(0)          .                                   01612
        LDA(3M)         SAL(2V)         . REWIND LISTABLE-OUTPUT MEDIUM     01613
        SLJ4(1V)        ZRO(0)          .                                   01614
```

```
        LDA(6M)          SAL(2V)          . REWIND PUNCH-CARD MEDIUM          01615
        SLJ4(1V)         ZRO(0)           .                                   01616
        LDA(3M)          INA(-1)          . SENSE INPUT-OUTPUT CONFLICTS      01617
        AJP(1P)          LDA(6M)          .                                   01620
        INA(-1)          AJP(1P)          .                                   01621
        LDA(2M)          EXF(53544B)      . TURN ON REMOTE MONITOR LIGHT      01622
        THS(9)           SLJ(L+4)         .                                   01623
        SUB(3M)          AJP(1P)          .                                   01624
        LDA(2M)          SUB(6M)          .                                   01625
        AJP(1P)          ENI(0)           .                                   01626
        ENI6(1)          SLJ4(10C)        . GO INITIALIZE FOR NEXT CALL       01627
        LDA(2K)          STA(T+9)         .                                   01630
        LDA(3M)          AJP(11M)         . SENSE PARTIAL MONITOR CONTROL     01631
        LDA(2M)          ENI3(0)          .                                   01632
        EQS(63B)         SLJ(L+6)         . SENSE STACKED-JOB DECK            01633
        EXF(62540B)      SLJ4(1G)         . GET NUMERIC DESIGNATOR            01634
        SIL2(T+11)       ENQ(M+63)        .                                   01635
        ENA2(14001B)     SAL(L+1)         .                                   01636
        STQ(Z+1)         EXF(N)           .                                   01637
        EXF1(M+39)       EXF7(24004B)     . READ BCD RECORD                   01640
        THS(9)           SLJ(L+13)        . SENSE STACKED-JOB TAPE            01641
        SLJ4(1R)         ZRO(0)           . GO READ BCD RECORD                01642
        ENA(M+39)        SAL(L+8)         . ASSEMBLE READ BUFFER              01643
        LDA3(R)          ALS(6)           .                                   01644
        ADD3(R+1)        ALS(6)           .                                   01645
        ADD3(R+2)        ALS(6)           .                                   01646
        ADD3(R+3)        ALS(6)           .                                   01647
        ADD3(R+4)        ALS(6)           .                                   01650
        ADD3(R+5)        ALS(6)           .                                   01651
        ADD3(R+6)        ALS(6)           .                                   01652
        ADD3(R+7)        STA(N)           . TRANSFER WORD TO MONITOR BUFFER   01653
        INI3(7)          RAO(L-1)         .                                   01654
        ISK3(79)         SLJ(L-9)         . SENSE TEN WORDS TRANSFERRED       01655
        SIL6(2C)         SIL6(R)          .                                   01656
        LDA(4M)          AJP1(11M)        . SENSE INTERMEDIATE COMPILE TAPE   01657
        LDA(R15)         STA(3C)          .                                   01660
        EXF(62560B)      SLJ4(1C)         . LOAD COMPILER                     01661
        SIL6(S+2)        ENA(0)           .                                   01662
        STA(3C)          SLJ(11M)         .                                   01663
9M      LDQ(2M)          QJP(2P)          . SUCCESS RETURN                    01664
10M     EXF(62560B)      EXF(53544B)      . RELEASE COMM FLAG 1               01665
        LDQ(2M)          QJP(1P)          . ERROR RETURN                      01666
        ENI6(1)          SLJ4(3T)         . GO OUTPUT JOB-TIME                01667
11M     EXF(62560B)      SLJ4(1J)         . GO RESET SWITCH SIMULATORS        01670
        ENI2(7)          EXF(53511B)      .                                   01671
        ENA(-0)          STA(1SWT)        . RE-SET 160 LIB CALL LOCK-OUT      01672
        ENI1(7)          ENA(0)           .                                   01673
        SAU2(C+8)        IJP2(L)          . RESET TAPE-ASSIGNMENT TABLE       01674
        LDA(3M)          AJP(L+2)         .                                   01675
```

```
        STA(4K)          ENI(0)          .                                  01676
        EQS1(C+9)        SLJ(L+2)        . SENSE LISTABLE-OUTPUT TAPE       01677
        ENA(-0)          SAU1(C+9)       . CANCEL LISTABLE-OUTPUT TAPE      01700
        LDA(2M)          ENI1(7)         .                                  01701
        EQS1(C+9)        SLJ(L+2)        . SENSE STACKED-JOB TAPE           01702
        ENA(-0)          SAU1(C+9)       . CANCEL STACKED-JOB TAPE          01703
        LDA(6M)          ENI1(7)         .                                  01704
        EQS1(C+9)        SLJ(L+2)        . SENSE PUNCH-CARD TAPE            01705
        ENA(-0)          SAU1(C+9)       . CANCEL PUNCH-CARD TAPE           01706
        ENA(0)           STA(CANCEL)     . CANCEL UNASSIGNED UNITS          01707
        ENA(32001B)      SAL(L+3)        .                                  01710
        SAU(L+4)         ENI1(1)         .                                  01711
        ENA(10B)         RAD(L+1)        .                                  01712
        ENI2(0)          EXF(32001B)     .                                  01713
        RAO(CANCEL)      ISK2(3000B)     .                                  01714
        EXF7(32001B)     SLJ(L+3)        .                                  01715
        ENA(77777B)      ENI(0)          .                                  01716
        LIL2(CANCEL)     SAU2(C+7)       .                                  01717
        ISK1(4B)         SLJ(L-6)        .                                  01720
        ENA(52001B)      SAU(L-9)        .                                  01721
        LIL2(CANCEL)     ENI(0)          .                                  01722
        ISK2(10B)        SLJ(L-11)       .                                  01723
        ENA(-0)          SAU(C+15)       . CANCEL GRAPH OUTPUT TAPE         01724
        ENA(32001B)      SAU(L-13)       .                                  01725
        EXF(70B)         LDA(R12)        . RESET TIME                       01726
        STA(Z)           ENA(0)          .                                  01727
        LDQ(4M)          QJP(L+2)        . INSURE INTERMEDIATE TAPE RESET   01730
        LIL1(4M)         SAU1(C+7)       .                                  01731
        STA(Z+50B)       EXF(100B)       .                                  01732
        EXF(1000B)       LDQ(5M)         .                                  01733
        QJP(L+2)         STQ(5T)         . SENSE JOB-TIME LIMIT             01734
        STA(6T)          SLJ4(4T)        . SET JOB-TIME LIMIT               01735
        ENI1(119)        ENQ(20B)        .                                  01736
        LDA1(R)          STA1(W+1)       . TRANSFER READ TO WRITE BUFFER    01737
        IJP1(L-1)        STQ(W)          .                                  01740
        ADD(R+1)         INA(-166B)      .                                  01741
11MAA   AJP(L+2)         SLJ4(Z+11B)     . SENSE JOB-DESCRIPTION RECORD     01742
        SLJ(L-5)         ZRO(0)          . RETURN FOR NEXT RECORD           01743
11MA    SSK(OSWT)        SLJ4(1SWTH)     . GO PROCESS SATELLITE JOB         01744
        ENA(-0)          STA(7M)         . CLEAR SOURCE INDICATOR           01745
        STA(8M)          ENI(0)          . CLEAR MAP INDICATOR              01746
        LDA(R+5)         INA(-54B)       . SENSE DESIRE SOURCE LIST         01747
        AJP1(L+1)        RAO(7M)         . SET SOURCE INDICATOR             01750
        LDA(R+6)         INA(-54B)       . SENSE DESIRE MAP LIST            01751
        AJP1(L+1)        RAO(8M)         . SET MAP INDICATOR                01752
        ENA(23B)         SLJ4(1W)        . TYPE JOB-DESCRIPTION RECORD      01753
        SIL6(2C)         SIL6(W)         .                                  01754
        LDA(3M)          AJP1(L+3)       . SENSE JOB-INCORPORATED CONTROL   01755
        LDA(3K)          SLJ4(1W)        .                                  01756
```

```
        SLJ(L+2)        ZRO(0)          .                                   01757
        SLJ4(Z+12B)     ZRO(0)          . STACK JOB-DESCRIPTION RECORD      01760
        LDA(6M)         AJP(12M)        . SENSE PUNCH-CARD MEDIUM           01761
        ENA(0)          SLJ4(3G)        . SENSE ALL CHANNELS INACTIVE       01762
        ENI1(119)       LDA(6M)         .                                   01763
        LDQ1(R)         STQ1(W)         . TRANSFER READ TO WRITE BUFFER     01764
        IJP1(L-1)       SLJ4(1W)        . STACK JOB-DESCRIPTION RECORD      01765
12M     LDA(R+2)        ALS(6)          .                                   01766
        ADD(R+3)        INA(-6545B)     .                                   01767
        AJP1(13M)       LDA(R+4)        .                                   01770
        INA(-64B)       AJP1(13M)       . SENSE END OF JOB-STACK            01771
        EXF(101B)       EXF(2000B)      . STOP CLOCK                        01772
        LDA(3M)         SAL(6V)         .                                   01773
        SLJ4(5V)        ZRO(0)          . ENDFILE LISTABLE-OUTPUT MEDIUM    01774
        LDA(6M)         SAL(6V)         .                                   01775
        SLJ4(5V)        ZRO(0)          . ENDFILE PUNCH-CARD MEDIUM         01776
        LDA(3M)         SAL(4V)         .                                   01777
        THS(9)          SLJ(L+2)        . SENSE LISTABLE-OUTPUT TAPE        02000
        SLJ4(3V)        ZRO(0)          . REWIND LISTABLE-OUTPUT TAPE       02001
        LDA(6M)         SAL(4V)         .                                   02002
        THS(9)          SLJ(L+2)        . SENSE PUNCH-CARD TAPE             02003
        SLJ4(3V)        ZRO(0)          . REWIND PUNCH-CARD TAPE            02004
        ENA(-0)         STA(2M)         . CANCEL MONITOR CONTROL            02005
        LDA(T+9)        STA(2K)         .                                   02006
        ENA(1)          SAL(2V)         . REWIND LIBRARY TAPE               02007
        SLJ4(1V)        ZRO(0)          .                                   02010
        LDA(R+5)        INA(31B)        .                                   02011
        EQS(65B)        SLJ(2P)         . SENSE SWITCH-BACK TO MONITOR      02012
        STA(R)          STA(R+1)        .                                   02013
        SLJ(3P)         ZRO(0)          . GO TO EXECUTE MONITOR             02014
13M     LDA(4M)         AJP1(15M)       . SENSE INTERMEDIATE COMPILE TAPE   02015
        LDA(3M)         AJP1(14M)       . SENSE JOB-INCORPORATED CONTROL    02016
        LDA(2M)         STA(2K)         .                                   02017
        ENA(-0)         STA(3M)         .                                   02020
        STA(6M)         ENI(0)          .                                   02021
        EXF(70B)        SLJ(2P)         . RETURN FOR CONTROL STATEMENT      02022
14M     LDA(7M)         STA(S+4)        . RUN IN FORTRAN-TO-MEMORY MODE     02023
        LDA(8M)         STA(S+6)        .                                   02024
        SLJ4(S)         ZRO(0)          . COMPILE TO MEMORY                 02025
        ENA(-0)         SAU(C+8)        . CANCEL LIBRARY TAPE               02026
        STA(1SWT)       EXF(53511B)     . RE-SET 160 LIB CALL LOCK-OUT      02027
        EXF(70B)        SLJ4(S+14000B)  . EXECUTE OBJECT PROGRAM            02030
        EXF(70B)        SLJ(9M)         . RETURN FOR NEXT JOB               02031
15M     LDA(R15)        STA(3C)         . RUN IN FORTRAN-TO-TAPE MODE       02032
        SLJ4(1C)        ZRO(0)          . LOAD COMPILER                     02033
        SIL6(S+2)       LDA(4M)         .                                   02034
        STA(S+3)        ENI1(7)         .                                   02035
        EQS1(C+9)       SLJ(1P)         .                                   02036
        SAU1(C+9)       ENI(0)          . CANCEL INTERMEDIATE TAPE          02037
```

```
        SLJ4(Z+16B)     ZRO(0)          . REWIND INTERMEDIATE TAPE          02040
        LDA(7M)         STA(S+4)        .                                   02041
        LDA(8M)         STA(S+6)        .                                   02042
        SLJ4(S)         ZRO(0)          . COMPILE TO TAPE                   02043
        LIL6(4M)        SIL6(2C)        .                                   02044
        ENA6(0)         SLJ4(Z+16B)     . REWIND INTERMEDIATE TAPE          02045
        ENA(0)          SAU6(C+7)       . RELEASE INTERMEDIATE TAPE         02046
        STA(3C)         SLJ4(1C)        . CALL OBJECT PROGRAM               02047
        ENA(-0)         SAU(C+8)        . CANCEL LIBRARY TAPE               02050
        STA(1SWT)       EXF(53511B)     . RE-SET 160 LIB CALL LOCK-OUT      02051
        EXF(70B)        SLJ4(S)         . EXECUTE OBJECT PROGRAM            02052
        EXF(70B)        SLJ(9M)         . RETURN FOR NEXT JOB               02053

* WRITE BINARY RECORD

10      SLJ(N)          INA(-2020B)     . EXIT/ENTRY                        02054
        AJP1(L+1)       ENA(63B)        . SENSE CARD REFERENCE              02055
        INA(2020B)      SLJ4(2G)        . GET NUMERIC DESIGNATOR            02056
        IJP1(30)        IJP2(2O)        .                                   02057
        SLJ(10)         ZRO(0)          . EXIT                              02060
20      ENA2(-5)        AJP1(5O)        . SENSE MEMORY REFERENCE            02061
        LDA(5S)         INA(-1)         . WRITE MEMORY RECORD               02062
        SAU(L+2)        INA(1)          .                                   02063
        LDQ1(W)         STQ7(5S)        .                                   02064
        ISK1(N)         SLJ(L-1)        .                                   02065
        RAD(6S)         SLJ(1O)         . EXIT                              02066
30      IJP1(L+1)       SLJ(4O)         . SENSE TAPE REFERENCE              02067
        LDA(U1)         ENI(0)          . WRITE TAPE RECORD                 02070
        IJP1(L+1)       LDA(UO)         .                                   02071
        IJP1(10)        ENI(0)          .                                   02072
        STA(T+10)       ENA(W+54)       .                                   02073
        SAU(3Y)         ENA2(0)         .                                   02074
        ALS(3)          INA(1)          .                                   02075
        ADD(T+10)       SLJ4(1Y)        .                                   02076
        SLJ(10)         ZRO(0)          . EXIT                              02077
40      ENA2(-2)        SLJ4(1FLAG)     . SENSE CARD REFERENCE              02100
        AJP1(50)        EXF7(24004B)    . SENSE CARD REFERENCE              02101
        ENI2(0)         ISK2(1000B)     . DELAY                             02102
        ENA(W+24)       ENI2(17)        . PUNCH BINARY CARD GROUP           02103
        EXF(24004B)     SAL(Z+2)        .                                   02104
        EXF2(W)         ENA(0)          . PUNCH CARD ONE                    02105
        STA2(W+54)      IJP2(L)         . CLEAR                             02106
        ENI2(0)         ISK2(1000B)     . DELAY                             02107
        ENI1(71)        ENI2(53)        .                                   02110
        ADD2(W)         IJP2(_)         . GET CHECK SUM                     02111
        EXF(70B)        STA1(W)         . STORE LOW ORDER PART              02112
        INI1(-1)        ENQ(-O)         .                                   02113
        LLS(8)          STQ1(W)         . STORE HIGH ORDER PART             02114
```

```
        ENA(W+48)           EXF7(24004B)        .                                   02115
        EXF(24004B)         SAL(Z+2)            .                                   02116
        EXF2(W+24)          ENI1(0)             . PUNCH CARD TWO                    02117
        ENA(W)              SAU(L+3)            . ASSEMBLE EXTRA BITS               02120
        ENI2(0)             ISK2(1000B)         . DELAY                             02121
        ENI2(0)             ENA(0)              .                                   02122
        LDQ2(N)             LLS(8)              .                                   02123
        ISK2(4)             SLJ(L-1)            .                                   02124
        STA1(W+54)          ENA(5000B)          .                                   02125
        ALS(15)             RAD(L-3)            .                                   02126
        ISK1(10)            SLJ(L-5)            .                                   02127
        LDA1(W+48)          STA1(M+15)          . TRANSFER TO BUFFER                02130
        ISK1(23)            SLJ(L-1)            .                                   02131
        ENA(M+39)           EXF7(24004B)        .                                   02132
        EXF(24004B)         SAL(Z+2)            .                                   02133
        EXF2(M+15)          EXF7(24004B)        . PUNCH CARD THREE                  02134
60      EXF(62560B)         SLJ(1D)             .                                   02135
50      ENA(E12)            SLJ(1E)             . ERROR EXIT                        02136

* PROGRAM CONTROL ROUTINE

1P      ENA(0)              STA(2K)             . SET CONTROL AT TYPEWRITER         02137
2P      EXF(53511B)         EXF(63544B)         . FORTRAN ON / MONITOR OFF          02140
        ENA(-0)             STA(1SWT)           . RE-SET 160 LIB CALL LOCK-OUT      02141
        LDA(2K)             AJP1(4P)            . SENSE TYPEWRITER CONTROL          02142
        EXF(2000B)          EXF(11100B)         . SELECT TYPEWRITER                 02143
        ENA(0)              ENI2(120)           .                                   02144
        STA2(R)             IJP2(L)             . CLEAR BUFFER                      02145
        ENA(T+1)            SAL(Z+1)            .                                   02146
        EXF1(T)             ISK1(100B)          . READ CHARACTER                    02147
        EXF7(11101B)        SLJ(2P)             . SENSE CARRIAGE RETURN             02150
        EXF7(11B)           SLJ(L-1)            . SENSE CHARACTER                   02151
        LIL1(T)             LDQ1(K0)            . TABLE LOOKUP                      02152
        EXF7(11141B)        QLS(6)              . SHIFT ON LOWER CASE               02153
        QLS(30)             LDL(77B)            .                                   02154
        STA2(R)             INA(-73B)           .                                   02155
        INI2(1)             AJP3(L-7)           . LOOP UNTIL PERIOD OR OPEN PAREN   02156
        EXF(1000B)          STA(3K)             .                                   02157
        STA(4K)             ENI(0)              .                                   02160
        ENA(20B)            STA2(R)             .                                   02161
        ISK2(120)           SLJ(L-1)            .                                   02162
3P      LDA(R11)            STA(Z+57B)          . NULLIFY LIMIT AT 57               02163
        ENA(-0)             ENI1(6)             . CANCEL MONITOR CONTROL            02164
        STA1(2M)            IJP1(L)             .                                   02165
        SLJ(L+2)            ZRO(0)              .                                   02166
4P      LDA(2K)             SLJ4(1R)            . READ CONTROL STATEMENT            02167
        ENI1(71)            ENI2(47)            .                                   02170
        LDA1(R)             STA1(W)             . COPY BUFFER                       02171
```

```
        IJP1(L-1)       ENA(20B)        .                                       02172
        STA2(W+72)      IJP2(_)         .                                       02173
        LDA(3K)         AJP5(1W)        . OUTPUT STATEMENT IF INDICATED         02174
        ENA(73B)        ENI4(0)         .                                       02175
        EQS(R)          SLJ(L+3)        . SENSE DOUBLE PERIOD                   02176
        EQS(R+1)        SLJ(4P)         .                                       02177
        SLJ(10M)        ZRO(0)          .                                       02200
        LDA4(R)         INA(-54B)       .                                       02201
        AJP1(L+1)       INI4(1)         . SENSE ASTERISK                        02202
        LDA4(R)         ENI1(26)        .                                       02203
        EQS1(F12)       SLJ(9M)         . SENSE LETTER                          02204
        ENA(0)          STA(T)          .                                       02205
        LDA4(R)         ENI1(36)        .                                       02206
        EQS1(F0)        SLJ(L+4)        . ASSEMBLE NAME                         02207
        LDA(T)          ALS(6)          .                                       02210
        ADD4(R)         STA(T)          .                                       02211
        INI4(1)         SLJ(L-4)        .                                       02212
        SUB(20B)        AJP(L-1)        .                                       02213
        LDA(T)          ENQ(0)          . POSITION NAME                         02214
        LRS(6)          AJP1(L)         .                                       02215
        LDL(-0B)        LIL3(L+5)       .                                       02216
        EQS3(P0)        SLJ(5P)         . SEARCH CALLED PROGRAMS                02217
        ENA3(-7)        AJP2(L+2)       . SENSE MONITOR-RESTRICTED GROUP        02220
        SSK(2M)         SLJ(2P)         . SENSE NO MONITOR CONTROL              02221
        LDA3(0Z)        SAL(L+3)        .                                       02222
        SAL(7P)         ARS(24)         .                                       02223
        STA(8P)         SLJ4(6P)        .                                       02224
        EXF(70B)        SLJ4(N)         . EXECUTE PROGRAM                       02225
        EXF(70B)        SLJ(2P)         . RETURN FOR CONTROL STATEMENT          02226
5P      EQS(R17)        SLJ(L+6)        . SENSE REMARK                          02227
        ENI1(79)        ENA(23B)        .                                       02230
        LDQ1(R)         STQ1(W)         .                                       02231
        IJP1(L-1)       SLJ4(1W)        . TYPE REMARK                           02232
        ENA(R5)         SLJ4(5G)        .                                       02233
        SLJ(2P)         ZRO(0)          . RETURN FOR CONTROL STATEMENT          02234
        ENA(E0)         SLJ(1E)         . ERROR EXIT                            02235
6P      SLJ(N)          SLJ(L+3)        . PROCESS ARGUMENTS                     02236
7P      ZRO(0)          ZRO(0)          . NEXT ARGUMENT ADDRESS                 02237
8P      ZRO(0)          ZRO(0)          . NUMBER OF ARGUMENTS                   02240
9P      LDA4(R)         AJP(11P)        .                                       02241
        INA(-33B)       AJP(L+5)        . SENSE COMMA OR OPEN PAREN             02242
        INA(-1)         AJP(L+4)        .                                       02243
        INA(-37B)       AJP(6P)         . SENSE PERIOD OR CLOSED PAREN          02244
        INA(-1)         AJP(6P)         .                                       02245
        SLJ(11P)        ZRO(0)          .                                       02246
        INI4(1)         LDA4(R)         . SKIP SPACES                           02247
        SUB(20B)        AJP(L-1)        .                                       02250
        ENA(0)          STA(T)          . CLEAR ASSEMBLY                        02251
        LDA4(R)         ENI1(3)         .                                       02252
```

```
        EQS1(F0)        SLJ(L+6)        . SENSE OCTAL NUMBER                    02253
        LDA4(R)         ENI1(8)         .                                       02254
        EQS1(F0)        SLJ(10P)        .                                       02255
        LDA(T)          ALS(3)          . ASSEMBLE NUMBER                       02256
        INA1(0)         STA(T)          .                                       02257
        INI4(1)         SLJ(L-4)        . LOOP                                  02260
        LDA4(R)         ENI1(36)        .                                       02261
        EQS1(F0)        SLJ(L+4)        .                                       02262
        LDA(T)          ALS(6)          . ASSEMBLE NAME                         02263
        ADD4(R)         STA(T)          .                                       02264
        INI4(1)         SLJ(L-4)        . LOOP                                  02265
        SUB(20B)        AJP(L-1)        .                                       02266
10P     RAO(7P)         RSO(8P)         .                                       02267
        AJP3(11P)       LDA(T)          . SENSE ARGUMENT LIMIT                  02270
        STA7(7P)        SLJ(9P)         . LOOP                                  02271
11P     ENA(E18)        SLJ(1E)         . ERROR EXIT                            02272

* EXECUTE LAST ASSEMBLED PROGRAM

1Q      ENI(0)          LIL1(C+5)       . ENTRY                                 02273
        INI1(-1)        LDA1(0Z)        .                                       02274
        SAU(L+1)        ENI(0)          .                                       02275
        SLJ4(N)         ZRO(0)          . EXECUTE PROGRAM                       02276
        SLJ(9M)         ZRO(0)          . RETURN TO MONITOR                     02277

* HOLD ROUTINE

2Q      SLJ(N)          LDA(C+5)        . EXIT/ENTRY                            02300
        STA(C+4)        LDA(C+3)        .                                       02301
        STA(C+2)        ENA(1)          .                                       02302
        STA(Z+47B)      SLJ(L-3)        . EXIT                                  02303

* CLEAR ROUTINE

3Q      SLJ(N)          SLJ4(10C)       . EXIT/ENTRY                            02304
        ENA(0)          STA1(S)         .                                       02305
        ISK1(77776B)    SLJ(L-1)        .                                       02306
        SLJ(L-3)        ZRO(0)          . EXIT                                  02307

* SKIP TO NEXT OR SPECIFIED JOB

4Q      SLJ(N)          SLJ(L+3)        . EXIT/ENTRY                            02310
        ZRO(0)          ZRO(0)          . MEDIUM                                02311
        ZRO(0)          ZRO(0)          . CHARACTERS IDENTIFYING JOB            02312
        LDA(L-2)        SLJ4(1R)        . READ RECORD                           02313
```

```
        LDA(R)          ADD(R+1)        .                                   02314
        INA(-166B)      AJP3(L-2)       . SENSE DOUBLE PERIOD               02315
        LDA(L-4)        AJP(L-6)        . SENSE SPECIFIC JOB REQUEST        02316
        ENI1(0)         ENQ(77B)        . ACCUMULATE CHARACTERS             02317
        ENA(0)          ENI(0)          .                                   02320
        ALS(6)          ADL1(R+2)       .                                   02321
        EQS(L-8)        SLJ(L+3)        . SENSE SPECIFIC JOB                02322
        ENA(0)          STA(L-9)        .                                   02323
        SLJ(L-12)       ZRO(0)          . EXIT                              02324
        ISK1(7)         SLJ(L-4)        . SENSE EIGHT CHARACTERS            02325
        SLJ(L-11)       ZRO(0)          . RETURN                            02326

* BCD READ ROUTINE

1R      SLJ(N)          SLJ4(1G)        . EXIT/ENTRY                        02327
        IJP1(L+10)      IJP2(L+6)       . SENSE NO MEDIUM REFERENCE         02330
        ENA(0)          ENI1(119)       . ENDFILE BLOCK TO BUFFER           02331
        STA1(R+1)       ENA(20B)        .                                   02332
        STA1(R)         IJP1(L)         .                                   02333
        ENA(73B)        STA(R)          . ENTER DOUBLE PERIOD               02334
        STA(R+1)        SLJ(1R)         . EXIT                              02335
        IJP2(L+1)       SLJ(6R)         . SENSE TYPEWRITER INPUT            02336
        IJP2(L+1)       SLJ(7R)         . SENSE PAPER TAPE INPUT            02337
        ENA2(-3)        AJP(8R)         . SENSE MEMORY INPUT                02340
        SLJ(5R)         ZRO(0)          .                                   02341
        IJP1(9R)        ENI(0)          . SENSE MAGNETIC TAPE INPUT         02342
        ENI(0)          ENA(14002B)     . SENSE CARD INPUT                  02343
        IJP2(L+1)       INA(-1)         .                                   02344
        SAU(L+1)        IJP2(5R)        .                                   02345
2R      EXF(N)          SLJ4(1FLAG)     . PROCESS CARD READER               02346
        ENA(R+104)      SAL(Z+1)        .                                   02347
        EXF1(R+80)      EXF7(14004B)    . READ CARD                         02350
        EXF(62560B)     SLJ4(3R)        . GO PROCESS CARD IMAGE             02351
        ENI(0)          ENI1(79)        .                                   02352
        LDA1(R)         ENI2(47)        .                                   02353
        EQS2(F0)        SLJ(L+2)        . SENSE VALID CODES                 02354
        IJP1(L-2)       SLJ(1R)         . EXIT                              02355
        EQS(14B)        SLJ(L+3)        . SENSE ALTERNATE DASH              02356
        ENA(40B)        STA1(R)         .                                   02357
        SLJ(L-3)        ZRO(0)          .                                   02360
        ENA(T3)         SLJ4(5G)        . TROUBLE EXIT                      02361
        AJP(1R)         SLJ(2R)         . SENSE FOR ACTION TO TAKE          02362
3R      SLJ(N)          ENI1(80)        . PROCESS CARD IMAGE                02363
        ENA(0)          ENI2(79)        . CLEAR BUFFER                      02364
        STA2(R)         IJP2(L)         .                                   02365
        ENA(60B)        SLJ4(4R)        . PROCESS ROWS                      02366
        ENA(40B)        SLJ4(4R)        .                                   02367
        ENA(20B)        SLJ4(4R)        .                                   02370
```

```
        ENA(1)          SLJ4(4R)        .                               02371
        ENA(2)          SLJ4(4R)        .                               02372
        ENA(3)          SLJ4(4R)        .                               02373
        ENA(4)          SLJ4(4R)        .                               02374
        ENA(5)          SLJ4(4R)        .                               02375
        ENA(6)          SLJ4(4R)        .                               02376
        ENA(7)          SLJ4(4R)        .                               02377
        ENA(8)          SLJ4(4R)        .                               02400
        ENA(9)          SLJ4(4R)        .                               02401
        ENA(20B)        ENI1(80)        .                               02402
        EQS1(R)         SLJ(L+3)        . CORRECT ZERO CODES            02403
        ENA(12B)        STA1(R)         .                               02404
        ENA(20B)        SLJ(L-2)        . LOOP                          02405
        ENA(0)          ENI1(80)        .                               02406
        EQS1(R)         SLJ(L+3)        . CORRECT BLANK CODES           02407
        ENA(20B)        STA1(R)         .                               02410
        ENA(0)          SLJ(L-2)        . LOOP                          02411
        ENI1(39)        ENA(20B)        . FILL OUT BLANKS               02412
        STA1(R+80)      IJP1(L)         .                               02413
        ENA(0)          STA(R+120)      .                               02414
        SLJ(3R)         ZRO(0)          . RETURN                        02415
4R      SLJ(N)          STA(T)          . ROW PROCESSOR                 02416
        LDA1(R)         ALS(8)          .                               02417
        STA1(R)         ENI2(39)        .                               02420
        SSH1(R)         SLJ(L+2)        .                               02421
        LDA(T)          RAD2(R)         .                               02422
        ENI(0)          IJP2(L-2)       .                               02423
        LDA1(R+1)       ALS(8)          .                               02424
        STA1(R+1)       ENI2(39)        .                               02425
        SSH1(R+1)       SLJ(L+2)        .                               02426
        LDA(T)          RAD2(R+40)      .                               02427
        ENI(0)          IJP2(L-2)       .                               02430
        INI1(2)         SLJ(4R)         . RETURN                        02431
5R      ENA(E9)         SLJ(1E)         . ERROR EXIT                    02432
6R      EXF(11100B)     SLJ4(1FLAG)     . PROCESS TYPEWRITER            02433
        ENA(T+1)        SAL(Z+1)        .                               02434
        ENI2(0)         ENI1(119)       .                               02435
        EXF1(T)         ENA(0)          .                               02436
        STA1(R+1)       ENA(20B)        .                               02437
        STA1(R)         IJP1(L)         . CLEAR BUFFER                  02440
        EXF1(T)         ISK1(100B)      .                               02441
        EXF7(11101B)    SLJ(L+7)        . SENSE CARRIAGE RETURN         02442
        EXF7(11B)       SLJ(L-1)        . LOOP TO CHARACTERS            02443
        LIL1(T)         LDQ1(<0)        . TABLE LOOKUP                  02444
        EXF7(11141B)    QLS(6)          . SHIFT ON LOWER CASE           02445
        QLS(30)         LDL(77B)        .                               02446
        STA2(R)         ENI(0)          . STORE IN BUFFER               02447
        ISK2(119)       SLJ(L-7)        . LOOP                          02450
        EXF1(T+1)       EXF(11100B)     .                               02451
```

```
        ENA2(0)          AJP(L-9)         . LOOP ON BLANK ENTRY                02452
        INI2(-1)         LDA2(R)          .                                    02453
        SUB(20B)         AJP1(L+2)        . SENSE TERMINAL SPACE               02454
        SLJ(L-16)        ZRO(0)           .                                    02455
        EXF(62560B)      ENI(0)           .                                    02456
        EXF(10000B)      SLJ(1R)          . EXIT                               02457
7R      ENI(0)           SLJ(L+2)         . PROCESS PAPER TAPE                 02460
        ZRO(0)           ZRO(0)           . CURRENT CASE                       02461
        ENA(0)           SLJ4(1FLAG)      .                                    02462
        EXF(11200B)      ENI2(119)        .                                    02463
        STA2(R+1)        ENA(20B)         .                                    02464
        STA2(R)          IJP2(L)          . CLEAR BUFFER                       02465
        ENA(T+1)         SAL(2+1)         .                                    02466
7RA     EXF1(T)          EXF7(11B)        . READ CHARACTER                     02467
        LDA(T)           AJP(L-1)         . SENSE BLANK CHARACTER              02470
        INA(-43B)        AJP(L-2)         . SENSE STOP CODE                    02471
        INA(-34B)        AJP(L-3)         . SENSE DELETE CODE                  02472
        LDA(T)           INA(-57B)        .                                    02473
        AJP(L+2)         INA(10B)         .                                    02474
        AJP1(L+2)        ENA(-0)          .                                    02475
        STA(L-13)        SLJ(L-7)         . LOOP                               02476
        INA(47B)         ENI(0)           .                                    02477
        EQS(51B)         SLJ(L+2)         .                                    02500
        ENI2(6)          SLJ(L-10)        . PROCESS TAB                        02501
        INA(-45B)        AJP(L+7)         . SENSE CARRIAGE RETURN              02502
        LIL1(T)          LDQ1(<0)         . TABLE LOOKUP                       02503
        SSK(L-19)        QLS(6)           . SHIFT ON LOWER CASE                02504
        QLS(42)          LDL(77B)         .                                    02505
        STA2(R)          INI2(1)          . STORE IN BUFFER                    02506
        ENA2(-81)        AJP(7R)          . RETURN ON BUFFER LIMIT             02507
        SLJ(L-17)        ZRO(0)           .                                    02510
        ENA2(0)          AJP(L-19)        . LOOP ON BLANK ENTRY                02511
        EXF(62560B)      SLJ(1R)          . EXIT                               02512
8R      ENI(0)           LDA(5S)          . PROCESS MEMORY                     02513
        INA(-1)          SAU(L+2)         .                                    02514
        LDQ7(6S)         STQ1(R)          .                                    02515
        ISK1(N)          SLJ(L-1)         .                                    02516
        INA(1)           RAD(6S)          .                                    02517
        SLJ4(3G)         ZRO(0)           . DISASSEMBLE READ BUFFER            02520
        EXF(62560B)      SLJ(1R)          . EXIT                               02521
9R      ENA(16X)         SAL(9X)          . PROCESS MAGNETIC TAPE              02522
        LDA(U1)          ENI(0)           .                                    02523
        IJP1(L+1)        LDA(UJ)          .                                    02524
10R     STA(T+10)        ENA(R+15)        .                                    02525
        IJP1(5R)         SAU(2X)          .                                    02526
        SAU(6X)          LDA(T+10)        .                                    02527
        ARS(24)          STA(T+10)        .                                    02530
        ENA2(0)          ALS(3)           .                                    02531
        INA(2)           ADD(T+10)        .                                    02532
```

```
        SLJ4(1X)        ZRO(0)          .                                   02533
        SLJ4(3G)        ZRO(0)          . DISASSEMBLE READ BUFFER           02534
        SLJ(1R)         ZRO(0)          . EXIT                              02535

* INPUT MEDIUM DESIGNATOR

1S      SLJ(N)          SLJ(L+3)        . EXIT/ENTRY                        02536
        ZRO(0)          ZRO(0)          . SYMBOL                            02537
        ZRO(0)          ZRO(0)          . DEVICE                            02540
        LIL1(L-2)       ENQ(77B)        .                                   02541
        LDL(L-2)        ALS(6)          .                                   02542
        ENQ(70077B)     SSU1(G0)        .                                   02543
        STA1(G0)        SLJ(1S)         . EXIT                              02544

* OUTPUT MEDIUM DESIGNATOR

2S      SLJ(N)          SLJ(L+3)        . EXIT/ENTRY                        02545
        ZRO(0)          ZRO(0)          . SYMBOL                            02546
        ZRO(0)          ZRO(0)          . DEVICE                            02547
        LIL1(L-2)       ENQ(77B)        .                                   02550
        LDL(L-2)        ENQ(77700B)     .                                   02551
        SSU1(G0)        STA1(G0)        .                                   02552
        SLJ(2S)         ZRO(0)          . EXIT                              02553

* SET START OF MEMORY FILE

3S      SLJ(N)          SLJ(L+4)        . EXIT/ENTRY                        02554
4S      ZRO(0)          ZRO(S)          . START OF MEMORY FILE              02555
5S      ZRO(0)          ZRO(0)          . MEMORY RECORD LENGTH              02556
6S      ZRO(0)          ZRO1(S)         . END OF MEMORY FILE                02557
        LDA(4S)         SAL(6S)         . SAVE START OF MEMORY FILE         02560
        LDA(5S)         AJP1(3S)        . SENSE RECORD LENGTH               02561
        ENA(15)         STA(5S)         . SET RECORD LENGTH TO FIFTEEN      02562
        SLJ(3S)         ZRO(0)          . EXIT                              02563

* CLOCK ROUTINE

1T      SLJ(N)          LDA(R11)        . EXIT/ENTRY                        02564
        STA(Z+57B)      LDA(R12)        . RESET LIMIT AT 57                 02565
        STA(Z)          ENA(0)          . SET ONE-SECOND COUNT              02566
        STA(Z+50B)      EXF(1000B)      . RESET TIME AT 50                  02567
        EXF(1000B)      SLJ(1T)         . EXIT                              02570

* STOP CLOCK ROUTINE
```

```
2T      SLJ(N)          EXF(1000B)      . EXIT/ENTRY                    02571
        EXF(2000B)      SLJ(2T)         . EXIT                          02572

* TIME ROUTINE

3T      SLJ(N)          SLJ4(7T)        . EXIT/ENTRY                    02573
        LDA(Z+50B)      ENQ(0)          .                               02574
        DVI(60)         STQ(T+10)       .                               02575
        SLJ4(L+8)       ZRO(0)          . GET MINUTES                   02576
        ADD(R13)        STA(E21)        .                               02577
        LDA(T+10)       SLJ4(L+6)       . GET SECONDS                   02600
        ADD(R14)        STA(E23)        .                               02601
        ENA(E21)        LDQ(2M)         .                               02602
        QJP(L+1)        LDQ(4K)         .                               02603
        QJP4(6G)        SLJ4(3E)        .                               02604
        SLJ(3T)         ZRO(0)          .                               02605
        SLJ(N)          ENI2(0)         . CONVERT NUMBER TO BCD         02606
        THS(100)        INI2(1)         .                               02607
        THS(10)         INI2(1)         .                               02610
        LDQ(R6)         STQ(T)          .                               02611
        ENQ(0)          DVI2(30)        .                               02612
        SAL(T+1)        LIL1(T+1)       .                               02613
        LDA(T)          ALS(6)          .                               02614
        SCL(77B)        ADD1(F0)        .                               02615
        STA(T)          LLS(48)         .                               02616
        IJP2(L-5)       LDA(T)          .                               02617
        SLJ(L-10)       ZRO(0)          . RETURN                        02620

* LIMIT ROUTINE

4T      SLJ(N)          SLJ(L+3)        . EXIT/ENTRY                    02621
5T      ZRO(0)          ZRO(0)          . MINUTES                       02622
6T      ZRO(0)          ZRO(0)          . SECONDS                       02623
        SLJ4(7T)        ZRO(0)          .                               02624
        LDA(5T)         MUI(60)         . COMPUTE AND SET LIMIT         02625
        ADD(6T)         ADD(Z+50B)      .                               02626
        STA(Z+57B)      SLJ(4T)         . EXIT                          02627

* START CLOCK IF NOT RUNNING

7T      SLJ(N)          ENI(0)          . EXIT/ENTRY                    02630
        ENI1(0)         ENQ(0)          .                               02631
        LDA(Z)          ISK1(40000B)    . WAIT                          02632
        EQS(Z)          SLJ(7T)         . SENSE CLOCK RUNNING           02633
        STQ(Z+50B)      LDA(R12)        . INITIALIZE TIME               02634
```

```
        STA(Z)          EXF(1000B)      . START CLOCK                       02635
        EXF(1000B)      SLJ(7T)         . EXIT                              02636

* REWIND SPECIFIED MEDIUM

1V      SLJ(N)          SLJ(L+2)        . EXIT/ENTRY                        02637
2V      ZRO(0)          ZRO(0)          . MEDIUM                            02640
        ENI1(0)         ISK1(3000B)     . DELAY                             02641
        LDA(2V)         SLJ4(1G)        . GET NUMERIC DESIGNATOR            02642
        IJP1(L+2)       IJP2(L+13)      . SENSE TAPE REFERENCE              02643
        SLJ(1VA)        ZRO(0)          . EXIT                              02644
        IJP1(L+1)       SLJ(1VA)        .                                   02645
        LDA(U1)         ENI(0)          . SET SELECT CODES                  02646
        IJP1(L+1)       LDA(U0)         .                                   02647
        IJP1(1VA)       ARS(24)         .                                   02650
        STA(T)          SAL(L+5)        .                                   02651
        ENA2(0)         ALS(3)          .                                   02652
        INA(2)          ADD(T)          .                                   02653
        SAU(L+2)        INA(3)          .                                   02654
        SAU(L+2)        SLJ4(1FLAG)     .                                   02655
        EXF(N)          EXF7(N)         . REWIND TAPE                       02656
        EXF(N)          SLJ(1VA)        . EXIT                              02657
        IJP2(L+1)       SLJ(1VA)        .                                   02660
        IJP2(L+10)      ENA(0)          . SENSE PAPER TAPE REFERENCE        02661
        STA(T+1)        ENA(T+2)        . PROCESS PAPER TAPE                02662
        SAL(Z+2)        SLJ4(1FLAG)     .                                   02663
        EXF(21210B)     ENI1(100)       . PUNCH LEADER                      02664
        EXF2(T+1)       EXF7(21B)       .                                   02665
        IJP1(L-1)       ENI(0)          . LOOP                              02666
        ENA(57B)        STA(T)          .                                   02667
        ENA(45B)        STA(T+1)        .                                   02670
        EXF2(T)         EXF7(21B)       . PUNCH CARRIAGE RETURN             02671
        SLJ(1VA)        ZRO(0)          . EXIT                              02672
        INI2(-3B)       IJP2(1VA)       . SENSE MEMORY REFERENCE            02673
        LDA(4S)         SAL(6S)         . REWIND MEMORY FILE                02674
1VA     EXF(62560B)     SLJ(1V)         .                                   02675

* REWIND TAPE WITH INTERLOCK

3V      SLJ(N)          SLJ(L+2)        . EXIT/ENTRY                        02676
4V      ZRO(0)          ZRO(0)          . MEDIUM                            02677
        ENI1(0)         ISK1(3000B)     . DELAY                             02700
        LDA(4V)         SLJ4(1G)        . GET NUMERIC DESIGNATOR            02701
        IJP1(L+1)       SLJ(3V)         . SENSE TAPE REFERENCE              02702
        IJP1(L+1)       SLJ(3V)         .                                   02703
        LDA(U1)         IJP1(L+2)       . SET SELECT CODES                  02704
        LDA(U0)         SLJ4(1FLAG)     .                                   02705
```

```
        ARS(24)          STA(T)          .                                  02706
        SAL(L+4)         ENA2(0)         .                                  02707
        ALS(3)           INA(2)          .                                  02710
        ADD(T)           SAU(L+2)        .                                  02711
        INA(5)           SAU(L+2)        .                                  02712
        EXF(N)           EXF7(N)         . REWIND WITH INTERLOCK            02713
        EXF(N)           ENI(0)          .                                  02714
        EXF(62560B)      SLJ(3V)         . EXIT                             02715

3VV     SLJ4(3V)         ZRO(0)          .                                  02716
        SLJ(Z+33B)       ZRO(0)          . EXIT                             02717

* WRITE ENDFILE MARKER ON SPECIFIED MEDIUM

5V      SLJ(N)           SLJ(L+2)        . EXIT/ENTRY                       02720
6V      ZRO(0)           ZRO(0)          . MEDIUM                           02721
        ENI1(0)          ISK1(3000B)     . DELAY                            02722
        LDA(6V)          SLJ4(2G)        . GET NUMERIC DESIGNATOR           02723
        IJP1(L+1)        SLJ(L+13)       . SENSE TAPE REFERENCE             02724
        IJP1(L+1)        SLJ(5V)         .                                  02725
        LDA(U1)          SLJ4(1FLAG)     . SET SELECT CODES                 02726
        IJP1(L+1)        LDA(U0)         .                                  02727
        IJP1(5V)         INA(3)          .                                  02730
        SAU(L+6)         SCL(777B)       .                                  02731
        SAL(L+4)         SAL(L+5)        .                                  02732
        SAL(T)           ENA2(0)         .                                  02733
        ALS(3)           INA(2)          .                                  02734
        ADD(T)           SAU(L+1)        .                                  02735
        EXF(N)           EXF7(N)         . WRITE TAPE ENDFILE               02736
        EXF(N)           EXF7(N)         .                                  02737
        EXF(62560B)      SLJ(5V)         . EXIT                             02740
        ENA2(-6)         AJP1(5V)        . SENSE MEMORY REFERENCE           02741
        LIL1(5S)         INI1(-1)        . WRITE MEMORY ENDFILE             02742
        LDQ(R4)          LDA(5S)         .                                  02743
        STQ7(6S)         IJP1(L)         .                                  02744
        RAD(6S)          SLJ(5V)         . EXIT                             02745

* BACKSPACE SPECIFIED MEDIUM

7V      SLJ(N)           SLJ(L+2)        . EXIT/ENTRY                       02746
8V      ZRO(0)           ZRO(0)          . MEDIUM                           02747
        ENI1(0)          ISK1(3000B)     . DELAY                            02750
        LDA(8V)          SLJ4(1G)        . GET NUMERIC DESIGNATOR           02751
        IJP1(L+2)        IJP2(L+15)      . SENSE TAPE REFERENCE             02752
        SLJ(7V)          ZRO(0)          .                                  02753
        IJP1(L+1)        SLJ(7V)         .                                  02754
        LDA(U1)          SLJ4(1FLAG)     . SET SELECT CODES                 02755
```

```
        IJP1(L+1)       LDA(UJ)         .                               02756
        ALS(24)         ENI(0)          .                               02757
        IJP1(7V)        INA(6)          .                               02760
        SAU(L+6)        SCL(777B)       .                               02761
        SAL(L+4)        SAL(L+5)        .                               02762
        SAL(T)          ENA2(J)         .                               02763
        ALS(3)          INA(2)          .                               02764
        ADD(T)          SAU(L+1)        .                               02765
        EXF(N)          EXF7(N)         . BACKSPACE TAPE                02766
        EXF(N)          EXF7(N)         .                               02767
        EXF(62560B)     SLJ(7V)         . EXIT                          02770
        ENA2(-5)        AJP1(7V)        . SENSE MEMORY REFERENCE        02771
        LDA(5S)         RSB(6S)         . BACKSPACE MEMORY              02772
        SLJ(7V)         ZRO(0)          . EXIT                          02773

* BCD WRITE ROUTINE

1W      SLJ(N)          SLJ4(2G)        . EXIT/ENTRY                    02774
        IJP1(L+6)       IJP2(L+2)       . SENSE NO MEDIUM REFERENCE     02775
        SLJ(1W)         ZRO(0)          . EXIT                          02776
        IJP2(L+1)       SLJ(6W)         . SENSE TYPEWRITER OUTPUT       02777
        IJP2(L+1)       SLJ(7W)         . SENSE PAPER TAPE OUTPUT       03000
        ENA2(-3)        AJP(9W)         . SENSE MEMORY OUTPUT           03001
        SLJ(5W)         ZRO(0)          .                               03002
        IJP1(10W)       ENI(0)          . SENSE MAGNETIC TAPE OUTPUT    03003
        ENA2(-2)        AJP(2W)         . SENSE CARD OUTPUT             03004
        SLJ(5W)         ZRO(0)          .                               03005
2W      SLJ4(3W)        ZRO(0)          . PROCESS CARD PUNCH            03006
        ENA(W+24)       SLJ4(1FLAG)     .                               03007
        EXF(24004B)     SAL(Z+2)        .                               03010
        EXF2(W)         EXF7(14004B)    .                               03011
        EXF(62560B)     SLJ(1W)         . EXIT                          03012
3W      SLJ(N)          ENI(0)          . FORM CARD IMAGE               03013
        ENA(1)          ALS(7)          .                               03014
        STA(T)          ENI1(J)         .                               03015
4W      ENQ(77B)        LDL1(W)         .                               03016
        STA(T+1)        LIL2(T+1)       .                               03017
        LDA2(G0)        ALS(6)          . TABLE LOOKUP                  03020
        ENI2(23)        STA(T+1)        .                               03021
        INI2(-1)        ENI(0)          .                               03022
        SSH(T+1)        SLJ(L+2)        . TAG CONTROL WORDS             03023
        LDA(T)          RAD2(W)         .                               03024
        ENI(0)          IJP2(L-3)       . LOOP                          03025
        ENQ(77B)        LDL1(W+40)      .                               03026
        STA(T+1)        LIL2(T+1)       .                               03027
        LDA2(G0)        ALS(6)          . TABLE LOOKUP                  03030
        ENI2(23)        STA(T+1)        .                               03031
        INI2(-1)        ENI(0)          .                               03032
```

```
        SSH(T+1)        SLJ(L+2)        . TAG CONTROL WORDS             03033
        LDA(T)          RAD2(W+1)       .                               03034
        ENI(0)          IJP2(L-3)       . LOOP                          03035
        LDA(T)          ALS(1)          .                               03036
        STA(T)          ENI(0)          . SHIFT MARK                    03037
        ISK1(39)        SLJ(4W)         . LOOP                          03040
        LDA1(W)         ARS(7)          .                               03041
        STA1(W)         ENI(0)          . POSITION RESULTS              03042
        ISK1(23)        SLJ(L-2)        . LOOP                          03043
        SLJ(3W)         ZRO(0)          . RETURN                        03044
5W      ENA(E12)        SLJ(1E)         . ERROR EXIT                    03045
6W      ENA(R5)         SLJ4(7G)        . PROCESS TYPEWRITER            03046
        EXF(21110B)     ENI(0)          .                               03047
        ENA(T+1)        SAL(Z+2)        .                               03050
        ENI2(119)       ENI(0)          .                               03051
        LDA2(W)         INA(-20B)       . SEARCH FOR TERMINAL BLANKS    03052
        AJP1(L+1)       IJP2(L-1)       .                               03053
        SIU2(L+13)      ENI2(0)         . STORE LENGTH OF ENTRY         03054
        LDA2(W)         STA(T)          .                               03055
        LIL1(T)         LDQ1(KO)        . TABLE LOOKUP                  03056
        QLS(6)          LDL(77B)        .                               03057
        AJP1(L+4)       ENI(0)          . JUMP ON UPPER CASE            03060
        EXF7(111418)    SLJ(L+6)        . JUMP ON LOWER CASE            03061
        ENA(57B)        STA(T)          .                               03062
        SLJ(L+3)        ZRO(0)          .                               03063
        EXF7(111408)    SLJ(L+3)        .                               03064
        ENA(47B)        STA(T)          .                               03065
        EXF2(T)         EXF7(21B)       . CARRIAGE SHIFT                03066
        QLS(6)          STQ(T)          .                               03067
        EXF2(T)         EXF7(21B)       . PRINT CHARACTER               03070
        ISK2(N)         SLJ(L-12)       . LOOP                          03071
        EXF7(111418)    SLJ(1W)         . SENSE CARRIAGE DOWN           03072
        ENA(57B)        STA(T)          . SHIFT CARRIAGE DOWN           03073
        EXF2(T)         EXF7(21B)       .                               03074
        SLJ(1W)         ZRO(0)          . EXIT                          03075
7W      ENI2(119)       SLJ(L+2)        . PROCESS PAPER TAPE            03076
        ZRO(0)          ZRO(0)          . CURRENT CASE                  03077
        ENA(T+1)        SLJ4(1FLAG)     .                               03100
        EXF(21210B)     SAL(Z+2)        .                               03101
        LDA2(W)         INA(-20B)       . SEARCH FOR TERMINAL BLANKS    03102
        AJP1(L+1)       IJP2(L-1)       .                               03103
        SIU2(L+18)      ENI2(0)         . STORE LENGTH OF ENTRY         03104
        LDA2(W)         STA(T)          .                               03105
        INA(-20B)       AJP1(L+2)       . SENSE SPACE                   03106
        ENA(4)          SLJ(L+13)       .                               03107
        LIL1(T)         LDQ1(KO)        . TABLE LOOKUP                  03110
        QLS(18)         LDL(77B)        .                               03111
        AJP1(L+5)       ENI(0)          . JUMP ON CARRIAGE UP           03112
        LDA(L-12)       AJP(L+8)        . JUMP ON CARRIAGE DOWN         03113
```

```
        ENA(0)          STA(L-13)       .                               03114
        ENA(57B)        STA(T)          .                               03115
        SLJ(L+4)        ZRO(0)          .                               03116
        LDA(L-16)       AJP1(L+4)       .                               03117
        ENA(77B)        STA(L-17)       .                               03120
        ENA(47B)        STA(T)          .                               03121
7WA     EXF2(T)         EXF7(21B)       . CARRIAGE SHIFT                03122
        QLS(6)          LDL(77B)        .                               03123
        STA(T)          ENI(0)          .                               03124
7WB     EXF2(T)         EXF7(21B)       . PUNCH CHARACTER               03125
        ISK2(N)         SLJ(L-17)       . LOOP                          03126
        ENA(45B)        STA(T)          .                               03127
7WC     EXF2(T)         EXF7(21B)       . PUNCH CARRIAGE RETURN         03130
        EXF(62560B)     SLJ(1W)         . EXIT                          03131
9W      SLJ4(4G)        ZRO(0)          . PROCESS MEMORY                03132
        LDA(5S)         INA(-1)         .                               03133
        SAU(L+2)        INA(1)          .                               03134
        LDQ1(W)         STQ7(6S)        .                               03135
        ISK1(N)         SLJ(L-1)        .                               03136
        RAD(6S)         SLJ(1W)         . EXIT                          03137
10W     LDA(U1)         ENI(0)          . PROCESS MAGNETIC TAPE         03140
        IJP1(L+1)       LDA(U0)         .                               03141
        IJP1(5W)        STA(T+10)       .                               03142
        SLJ4(4G)        ZRO(0)          . ASSEMBLE WRITE BUFFER         03143
        ENA(W+15)       SAU(3Y)         .                               03144
        ENA2(0)         ALS(3)          .                               03145
        ADD(T+10)       INA(2)          .                               03146
        SLJ4(1Y)        ZRO(0)          .                               03147
        SLJ(1W)         ZRO(0)          . EXIT                          03150

* TAPE READING ROUTINE

1X      SLJ(N)          SLJ4(1FLAG)     . EXIT/ENTRY                    03151
        ENI(0)          SAU(4X)         .                               03152
        SCL(777B)       SAL(4X)         . INITIALIZE                    03153
        SAL(5X)         SAL(11X)        .                               03154
        INA(3)          SAU(8X)         .                               03155
        INA(2)          SAU(9X)         .                               03156
        INA(1)          SAU(11X)        .                               03157
        INA(1)          SAU(7K)         .                               03160
        LRS(18)         ENA(0)          .                               03161
        LLS(3)          SAL(2X)         .                               03162
        SAL(3X)         SAU(12X)        .                               03163
        LLS(39)         LDQ(R16)        .                               03164
        STA(T+10)       LDA(5X)         .                               03165
        SSU(T+10)       STA(5X)         .                               03166
        LDA(6X)         SSU(T+10)       .                               03167
        STA(6X)         ENI1(4)         .                               03170
```

```
        LDA(R6)          ENI(0)          .                                  03171
        STA1(R+10)       IJP1(L)         .                                  03172
2X      ENA(N)           SAL(N)          .                                  03173
3X      ENI2(4)          LIL1(N)         .                                  03174
4X      EXF(N)           EXF7(N)         . SELECT UNIT                      03175
5X      EXF(R)           EXF7(N)         . ACTIVATE CHANNEL                 03176
6X      EXF(N)           LDA(R)          .                                  03177
        SUB(R4)          LLS(48)         .                                  03200
        INI1(-1)         LDA1(Z)         .                                  03201
        INI1(1)          SUB(R4)         .                                  03202
7X      EXF7(N)          SLJ(15X)        . SENSE ENDFILE                    03203
8X      EXF7(N)          SLJ(L+4)        . SENSE PARITY ERROR               03204
9X      EXF7(N)          SLJ(10X)        . SENSE LENGTH ERROR               03205
        QJP1(1XA)        AJP(3X)         . SENSE FOR BAD RECORD             03206
        SLJ(1XA)         ZRO(0)          . EXIT                             03207
        IJP2(L+2)        SLJ(14X)        . SENSE FIVE PARITY ERRORS         03210
10X     IJP2(L+1)        SLJ(13X)        . SENSE FIVE LENGTH ERRORS         03211
        AJP(3X)          QJP(3X)         . SENSE FOR BAD RECORD             03212
11X     EXF(N)           EXF7(N)         . BACKSPACE                        03213
12X     LIL1(N)          SLJ(4X)         . RETURN TO REREAD                 03214
13X     ENA(T6)          SLJ(17X)        . SET TO INDICATE TROUBLE          03215
14X     ENA(T9)          SLJ(17X)        .                                  03216
15X     ENA(T12)         SLJ(13X)        .                                  03217
16X     ENA(10X)         SAL(9X)         . SET TO REREAD NEW LENGTH         03220
        LIU1(2X)         ENA(R+10)       .                                  03221
        SIL1(L-1)        SAL7(2X)        .                                  03222
        SAU(6X)          SAL(10R)        .                                  03223
        SLJ(11X)         ZRO(0)          . RETURN TO REREAD                 03224
17X     SLJ4(5G)         ZRO(0)          . TROUBLE EXIT                     03225
        AJP(1XA)         AJP3(3X)        . SENSE ACTION TO TAKE             03226
        SLJ(11X)         ZRO(0)          .                                  03227
18X     SLJ4(5G)         ZRO(0)          . TROUBLE EXIT                     03230
        AJP(3X)          AJP3(3X)        . SENSE ACTION TO TAKE             03231
        SLJ(11X)         ZRO(0)          .                                  03232
1XA     EXF(62560B)      SLJ(1X)         .                                  03233

* TAPE WRITING ROUTINE

1Y      SLJ(N)           SLJ4(1FLAG)     . EXIT/ENTRY                       03234
        ENI(0)           SAU(2Y)         .                                  03235
        SAU(9Y)          SCL(777B)       . INITIALIZE                       03236
        SAL(2Y)          SAL(4Y)         .                                  03237
        SAL(8Y)          SAL(10Y)        .                                  03240
        INA(3)           SAU(5Y)         .                                  03241
        SAU(11Y)         INA(2)          .                                  03242
        SAU(6Y)          SAU(12Y)        .                                  03243
        INA(1)           SAU(8Y)         .                                  03244
        SAU(7Y)          SAU(13Y)        .                                  03245
```

```
        LRS(15)          SAL(3Y)          .                                  03246
        SAL(15Y)         ENQ(0)           .                                  03247
        LLS(39)          LDQ(R16)         .                                  03250
        STA(T+10)        LDA(4Y)          .                                  03251
        SSU(T+10)        STA(4Y)          .                                  03252
        LDA(10Y)         SSU(T+10)        .                                  03253
        STA(10Y)         ENI(0)           .                                  03254
2Y      EXF(N)           EXF7(N)          . SELECT UNIT                      03255
3Y      ENA(N)           SAL(N)           .                                  03256
4Y      EXF(W)           EXF7(N)          . ACTIVATE CHANNEL                 03257
5Y      EXF7(N)          SLJ(17Y)         . SENSE PARITY ERROR               03260
6Y      EXF7(N)          SLJ(19Y)         . SENSE LENGTH ERROR               03261
7Y      EXF7(N)          SLJ(1YA)         . SENSE END OF TAPE                03262
        SLJ(16Y)         ZRO(0)           .                                  03263
8Y      EXF(N)           EXF7(N)          . BACKSPACE                        03264
9Y      EXF(N)           SLJ(14Y)         . SELECT UNIT                      03265
10Y     EXF(W+64)        EXF7(N)          . WRITE BAD RECORD INDICATION      03266
11Y     EXF7(N)          SLJ(2Y)          . SENSE PARITY ERROR               03267
12Y     EXF7(N)          SLJ(2Y)          . SENSE LENGTH ERROR               03270
13Y     EXF7(N)          SLJ(2Y)          . SENSE END OF TAPE                03271
        SLJ(16Y)         ZRO(0)           .                                  03272
14Y     ENI1(53)         LDA(R4)          .                                  03273
        STA1(W+64)       IJP1(L)          . SET BAD RECORD INDICATION        03274
15Y     ENA(64)          RAD(N)           .                                  03275
        IJP2(10Y)        SLJ(3Y)          .                                  03276
16Y     ENA(T21)         SLJ4(5G)         . TROUBLE EXIT                     03277
        AJP(2Y)          ENI2(0)          . SENSE ACTION TO TAKE             03300
        AJP3(1YA)        SLJ(8Y)          .                                  03301
17Y     ENA(T18)         SLJ4(5G)         . TROUBLE EXIT                     03302
18Y     AJP3(1YA)        ENI2(1)          . SENSE ACTION TO TAKE             03303
        AJP(8Y)          ENI2(0)          .                                  03304
        SLJ(8Y)          ZRO(0)           .                                  03305
19Y     ENA(T15)         SLJ4(5G)         . TROUBLE EXIT                     03306
        SLJ(18Y)         ZRO(0)           .                                  03307
1YA     EXF(62560B)      SLJ(1Y)          .                                  03310

* SERVICE ROUTINE CALL SERVICE

1SR     SLJ(N)           ENI1(0)          .                                  03311
        ENA(1)           STA(2C)          .                                  03312
        LDA1(COPYS)      STA1(3C)         .                                  03313
        ISK1(7B)         SLJ(L-1)         .                                  03314
        SLJ4(1C)         ZRO(0)           .                                  03315
        SLJ(1SR)         ZRO(0)           .                                  03316

* INTERRUPT HANDLER
```

```
1SENS  EXF7(32505B)      SLJ(3STOO)      . 160-3/4 INTERRUPT                    03317
       EXF7(52505B)      SLJ(5STOO)      . 160-5/6 INTERRUPT                    03320
       EXF7(00111B)      SLJ(1CLR)       . DIVIDE FAULT                         03321
       EXF7(00121B)      SLJ(1CLR)       . SHIFT FAULT                          03322
       EXF7(00131B)      SLJ(1A)         . OVERFLOW FAULT                       03323
       EXF7(00141B)      SLJ(1CLR)       . EXPONENT FAULT                       03324
       SLJ(L+1)          SLJ(N)          . CR INTERRUPT                         03325
       SLJ(L+1)          SLJ(N)          . CH 3 READY READ                      03326
       SLJ(L+1)          SLJ(N)          . CH 5 READY READ                      03327
       SLJ(L+1)          SLJ(N)          . CH 4 READY WRITE                     03330
       SLJ(L+1)          SLJ(N)          . CH 6 READY WRITE                     03331
       SLJ(L+1)          SLJ(N)          . CARD READER/PUNCH READY              03332
       SLJ(L+1)          SLJ(N)          . PAPER TAPE READER END OF TAPE        03333
       SLJ(L+1)          SLJ(N)          . DD65 DISPLAY INTERRUPT               03334
       SIU1(L+2)         LIU1(Z+7)       . ERROR INTERRUPT                      03335
       SIL1(L+1)         ENI(0)          .   LOCK-OUT                           03336
       ENI1(N)           SLJ(N)          . RETURN TO PROGRAM                    03337

1CLR   EXF(70B)          SLJ(Z+7)        . CLEAR ARITHMETIC FAULT               03340
2CLR   EXF(11100B)       SLJ(Z+7)        . CLEAR CR INTERRUPT                   03341

* CHANNEL 3-4 SATELLITE INTERRUPT ROUTINE

3STOO  STA(3TEMP)        STQ(3QTMP)      .SAVE A/Q REGISTERS                    03342
       SIU3(3PX22)       EXF(32505B)     .SAVE B3 / REL. INTERRUPT              03343
       LDA(3IBC)         STA(3STBF)      .SET 1 WORD INPUT CONTROL              03344
       ENA(3STCC)        SAL(3STBB)      .SET LOOP SWITCH                       03345
3STAA  EXF(32503B)       EXF7(32000B)    .SEL 160-1604 / WAIT READY             03346
       SIU1(3PXIT)       EXF7(42000B)    .WAIT WRITE READY                      03347
       LDA(3STBF)        SAL(Z+3)        .SET TERMINAL ADDRESS                  03350
       ARS(30B)          SAU(L+2)        .SET INITIAL ADDRESS                   03351
       EXF(32001B)       EXF(42504B)     .READ BINARY / 160 ACTION REQ.         03352
       EXF3(N)           EXF7(31B)       .ACTIVATE / WAIT CH INACTIVE           03353
       ENI(0)            EXF7(32000B)    .PASS / WAIT READY TO READ             03354
       ENA(3EXIT)        EXF(42500B)     .SET FOR LOOP / REL DIRECT             03355
3STBB  SIL2(3PXIT)       SLJ(N)          .SWITCH                                03356
3STCC  EXF7(42521B)      SLJ(3STDD)      .EXIT NO F2 / GO CASE S.R.             03357
       SSK(3STBF)        SLJ(L+2)        .INPUT OR OUTPUT / GO OUTPUT           03360
       SAL(3STBB)        SLJ(3STAA)      .SWITCH TO EXIT / GO INPUT             03361
3STCD  EXF(42503B)       EXF7(42000B)    .SEL 1604-160 / WAIT READY             03362
       ENI(0)            EXF7(32000B)    .WAIT READY READ                       03363
       LDA(3STBF)        SAL(Z+4)        .SET TERMINAL ADDRESS                  03364
       ARS(30B)          SAU(L+2)        .SET INITIAL ADDRESS                   03365
       EXF(42001B)       EXF(42504B)     .WRITE BINARY / 160 ACTION REQ.        03366
       EXF4(N)           EXF7(41B)       .ACTIVATE / WAIT CH INACTIVE           03367
       ENI(0)            EXF7(42000B)    .PASS / WAIT READY TO WRITE            03370
3EXIT  EXF(42500B)       EXF(42520B)     .REL DIRECT /REL F2                    03371
       EXF(32502B)       EXF(42502B)     .TAKE READ AND WRITE CONTROL           03372
```

```
3PXIT  ENI1(N)          ENI2(N)          .RESTORE B1 AND B2                    03373
3PX22  ENI3(N)          LDA(3TEMP)       .RESTORE B3 / A REG.                  03374
       LDQ(3QTMP)       SLJ(Z+7)         .RESTORE Q REG / EXIT VIA 7           03375
3STDD  LDA(3STBF)       SAL(L+7)         .PRIME FOR ARG INPUT                  03376
       SAL(3STED)       ENI1(1)          .    AND S.R. RETURN JUMP             03377
       SSK(3STBF)       SLJ(3STEE)       .CHECK ARG MODE / GO LDA ARG.         03400
       ALS(3B)          ARS(33B)         .PACK                                 03401
       INA(-1B)         ENI(0)           .    THE                              03402
       SAU(L+2)         ARS(17B)         .        ARGUMENT                     03403
       AJP(3STEE)       SAU(L+2)         .            ADDRESSES                03404
       ENA1(N)          STA1(N)          .                INTO                 03405
       ISK1(N)          SLJ(L-1)         .                    SUBROUTINE       03406
3STEE  ALS(3B)          ARS(33B)         .POSITION A REGISTER ARGUMENT         03407
3STED  ENI(0)           SLJ4(N)          .GO EXECUTE SUBROUTINE                03410
       SLJ(3EXIT)       ZRO(0)           .GO EXIT FROM INTERUPT                03411
3IBC   ZRO(3STBF)       ZRO(3TEMP)       .INITIAL BUFFER CONTROL               03412
3STBF  ZRO(0)           ZRO(0)           .BUFFER CELL                          03413
3TEMP  ZRO(0)           ZRO(0)           .TEMP STORAGE                         03414
3QTMP  ZRO(0)           ZRO(0)           .TEMP STORAGE FOR Q REGISTER          03415

* CHANNEL 5-6 SATELLITE INTERRUPT ROUTINE

5STOO  STA(5TEMP)       STQ(5QTMP)       .SAVE A/Q REGISTERS                   03416
       SIU3(5PX22)      EXF(52505B)      .SAVE B3 / REL. INTERRUPT             03417
       LDA(5IBC)        STA(5STBF)       .SET 1 WORD INPUT CONTROL             03420
       ENA(5STCC)       SAL(5STBB)       .SET LOOP SWITCH                      03421
5STAA  EXF(52503B)      EXF7(52000B)     .SEL 160-1604 / WAIT READY            03422
       SIU1(5PXIT)      EXF7(62000B)     .WAIT READY WRITE                     03423
       LDA(5STBF)       SAL(Z+5)         .SET TERMINAL ADDRESS                 03424
       ARS(30B)         SAU(L+2)         .SET INITIAL ADDRESS                  03425
       EXF(52001B)      EXF(62504B)      .READ BINARY / 160 ACTION REQ.        03426
       EXF5(N)          EXF7(51B)        .ACTIVATE / WAIT CH INACTIVE          03427
       ENI(0)           EXF7(52000B)     .PASS / WAIT CH INACTIVE              03430
       ENA(5EXIT)       EXF(62500B)      .SET FOR LOOP / REL DIRECT            03431
5STBB  SIL2(5PXIT)      SLJ(N)           .SWITCH                               03432
5STCC  EXF7(62521B)     SLJ(5STDD)       .EXIT NO F2 / GO CASE S.R.            03433
       SSK(5STBF)       SLJ(L+2)         .INPUT OR OUTPUT / GO OUTPUT          03434
       SAL(5STBB)       SLJ(5STAA)       .SWITCH TO EXIT / GO INPUT            03435
5STCD  EXF(62503B)      EXF7(52000B)     .SEL 1604-160 / WAIT READY            03436
       ENI(0)           EXF7(52000B)     .WAIT READY READ                      03437
       LDA(5STBF)       SAL(Z+6)         .SET TERMINAL ADDRESS                 03440
       ARS(30B)         SAU(L+2)         .SET INITIAL ADDRESS                  03441
       EXF(62001B)      EXF(62504B)      .WRITE BINARY / 160 ACTION REQ.       03442
       EXF6(N)          EXF7(61B)        .ACTIVATE / WAIT CH INACTIVE          03443
       ENI(0)           EXF7(52000B)     .PASS / WAIT READY TO WRITE           03444
5EXIT  EXF(62500B)      EXF(62520B)      .REL DIRECT / REL F2                  03445
       EXF(52502B)      EXF(62502B)      .TAKE READ AND WRITE CONTROL          03446
5PXIT  ENI1(N)          ENI2(N)          .RESTORE B1 AND B2                    03447
```

```
5PX22  ENI3(N)          LDA(5TEMP)      .RESTORE B3 / A REG.                    03450
       LDQ(5QTMP)       SLJ(Z+7)        .RESTORE Q REG / EXIT VIA 7             03451
5STDD  LDA(5STBF)       SAL(L+7)        .PRIME FOR ARG INPUT                    03452
       SAL(5STED)       ENI1(1)         .   AND S.R. RETURN JJMP                03453
       SSK(5STBF)       SLJ(5STEE)      .CHECK ARG MODE / GO LDA ARG.           03454
       ALS(3B)          ARS(33B)        .PACK                                   03455
       INA(-1B)         ENI(0)          .   THE                                 03456
       SAU(L+2)         ARS(17B)        .      ARGUMENT                         03457
       AJP(5STEE)       SAU(L+2)        .         ADDRESSES                     03460
       ENA1(N)          STA1(N)         .            INTO                       03461
       ISK1(N)          SLJ(L-1)        .               SUBROUTINE              03462
5STEE  ALS(3B)          ARS(33B)        .POSITION A REGISTER ARGUMENT           03463
5STED  ENI(0)           SLJ4(N)         .GO EXECUTE SUBROUTINE                  03464
       SLJ(5EXIT)       ZRO(0)          .GO EXIT FROM INTERUPT                  03465
5IBC   ZRO(5STBF)       ZRO(5TEMP)      .INITIAL BUFFER CONTROL                 03466
5STBF  ZRO(0)           ZRO(0)          .BUFFER CELL                            03467
5TEMP  ZRO(0)           ZRO(0)          .TEMP STORAGE                           03470
5QTMP  ZRO(0)           ZRO(0)          .TEMP STORAGE FOR Q REGISTER            03471

* 160 SATELLITE LIBRARY CALL ROUTINE HANDLER

160CL  STA(2NAME)       EXF(63511B)     . STORE NAME / TURN OFF LIGHT 1         03472
       LDA(BLOCKS)      STA(160TP)      . SET BLOCK READ LIMIT                  03473
       LDA(2X)          STA(2XSTR)      . PROTECT RESIDENT                      03474
       LDA(5X)          STA(5XSTR)      .     BINARY READ                       03475
       LDA(6X)          STA(6XSTR)      .         ROUTINE                       03476
       LDA(2NAME)       ENI1(10B)       . SEARCH IDENT TABLE                    03477
       EQS1(IDENT)      SLJ(1LAST)      .    TO IDENTIFY PROGRAM                03500
       LDA1(SIZE)       SAU(L+2)        . SET NR OF WORDS TO BE READ            03501
       SSK(1SWT)        SLJ(1LAST)      . EXIT IF BINARY READ LOCKOUT           03502
       ENI3(N)          LDA(2V)         . SAVE REWIND ARGUMENT                  03503
       STA(2VSTR)       ENA(13)         .                                       03504
       SAL(2V)          SLJ4(1V)        . REWIND LIBRARY TAPE                   03505
       LDA(2VSTR)       STA(2V)         . RESTORE REWIND ARG.                   03506
160LA  ENI(0)           SLJ4(160IN)     . GO READ LIB BLOCK                     03507
       RSO(160TP)       AJP(1LAST)      . CHECK FOR LIMIT                       03510
       SSK(2VSTR)       SLJ(1SRCH)      . CHECK HIT FLAG                        03511
160LB  LDA1(SATBUF)     STA(2NAME)      . SET WORD IN OUTPUT BUFFER             03512
       IJP3(160UT)      SLJ(1LAST)      . CHECK IF LAST WORD                    03513
160LC  ISK1(65B)        SLJ(L-2)        . LOOP UNTIL BLOCK IS EXHAUSTED         03514
       SLJ(160LA)       ZRO(0)          . GO READ NEXT BLOCK                    03515
160UT  SLJ4(160WD)      ZRO(0)          . GO WRITE OUT WORD                     03516
       SLJ(160LC)       ZRO(0)          . RETURN TO READ NEXT WORD              03517
160WD  SLJ(N)           ENA(2XSTR)      . WRITE OUT WORD ROUTINE                03520
       EXF(62503B)      EXF7(62000B)    .                                       03521
       SAL(Z+6)         EXF7(52000B)    .                                       03522
       EXF(62001B)      EXF(62504B)     .                                       03523
       EXF6(2NAME)      EXF7(51B)       .                                       03524
```

```
        ENI(0)          EXF7(62000B)    .                                   03525
        EXF(62560B)     SLJ(160WD)      .                                   03526
160IN   SLJ(N)          ENA(SATBUF)     . LIB BLOCK READ ROUTINE            03527
        SAU(5X)         INA(65B)        .                                   03530
        SAU(2X)         SAU(6X)         .                                   03531
        LDA(TAPE1)      SLJ4(1X)        .                                   03532
        ENI1(0)         SLJ(160IN)      .                                   03533
1SRCH   LDA(SATBUF)     AJP(160LA)      . SEARCH TAPE FOR NAME              03534
        SCM(2NAME)      AJP1(160LA)     . LOOP UNTIL HIT                    03535
        ENI1(1)         ENA(-0)         . SET HIT FLAG                      03536
        STA(2VSTR)      SLJ(150LB)      . GO TRANSMIT LIBRARY ROUTINE       03537
1LAST   EXF(62540B)     SLJ4(160WD)     . GO WRITE OUT LAST WORD            03540
        LDA(2XSTR)      STA(2X)         . RESTORE                           03541
        LDA(5XSTR)      STA(5X)         .     TAPE READ                     03542
        LDA(6XSTR)      STA(6X)         .         ROUTINE                   03543
        EXF(53511B)     EXF(62560B)     . TURN ON LIGHT 1 / REL COMM 1      03544
        SSK(1SWT)       SLJ(1CALL)      . EXIT, LOCK-OUT SET                03545
        EXF(32005B)     SLJ(1CALL)      . REWIND LIB / NORMAL EXIT          03546
2NAME   ZRO(0)          ZRO(0)          .                                   03547
2XSTR   ZRO(0)          ZRO(0)          .                                   03550
5XSTR   ZRO(0)          ZRO(0)          .                                   03551
6XSTR   ZRO(0)          ZRO(0)          .                                   03552
2VSTR   ZRO(0)          ZRO(0)          .                                   03553
160TP   ZRO(0)          ZRO(0)          .                                   03554

* SATELLITE MONITOR JOB ROUTINE HANDLER

1SWTH   SLJ(N)          LDA(2M)         . SAVE ON-LINE                      03555
        SAL(8V)         SLJ4(7V)        .    MONITOR JOB                    03556
        ENI(0)          SLJ4(7V)        .       DESCRIPTION                 03557
        ENI(0)          ENQ(7777B)      .                                   03560
        LDA(2M)         STA(2MSTR)      . SAVE MONITOR                      03561
        LDA(3M)         STA(3MSTR)      .     INPUT/OUTPUT                  03562
        LDA(4K)         STA(4KSTR)      .         ARGUMENTS                 03563
        ENA(2INTM)      SAL(Z+25B)      . SET FOR ERRORS TO SATELLITE       03564
        ENA(66B)        STA(4K)         . SET SAT FLEX ERROR OUTPUT         03565
        ENA(1INTM)      SAL(11MA)       . SET SWITCH DIVERTER               03566
        LDL(0SWT)       ENQ(65B)        . CHECK IF DESIRE PUNCH OUTPUT      03567
        AJP(L+1)        STQ(3M)         . IF SO, SET OUTPUT ARG             03570
        STQ(2M)         LDA(2SWTH)      . SET FLEX INPUT                    03571
        STA(7RA)        LDA(3SWTH)      . PRIME FLEX READ                   03572
        STA(7WA)        STA(7WB)        . PRIME FLEX WRITE                  03573
        STA(7WC)        ENA(2RCVR)      .                                   03574
        SAU(Z+30B)      ENA(11MA)       . SET TROUBLE RECOVERY              03575
        INA(-2B)        SAU(1SWTH)      . SET TO READ SAT JOB               03576
        SLJ(1SWTH)      ZRO(0)          . RETURN                            03577
2SWTH   SLJ4(1STRD)     ZRO(0)          . JUMP TO READ FLEX                 03600
3SWTH   SLJ4(1STWT)     ZRO(0)          . JUMP TO WRITE FLEX                03601
```

```
2MSTR  ZRO(0)           ZRO(0)          .                                    03602
3MSTR  ZRO(0)           ZRO(0)          .                                    03603
4KSTR  ZRO(0)           ZRO(0)          .                                    03604
1INTM  SLJ(N)           ENA(1RCVR)      . SET SWITCH TO RECOVER              03605
       SAL(11MA)        SLJ(L-1)        .                                    03606
2INTM  ENA(-0)          STA(1ERSW)      .                                    03607
       SLJ(Z+25B)       ZRO(0)          .                                    03610
1STRD  SLJ(N)           LDA(2STRD)      . SATELLITE READ ROUTINE             03611
       STA(5STBF)       LDA(L-1)        .                                    03612
       ALS(24)          EXF(2000B)      .                                    03613
       SAL(3STRD)       ENA(3STRD)      .                                    03614
       SAU(Z+7)         ENA(5EXIT)      .                                    03615
       SAL(5STBB)       SIU1(5PXIT)     .                                    03616
       SIL2(5PXIT)      SIU3(5PX22)     .                                    03617
       STQ(5QTMP)       ENI(0)          .                                    03620
       EXF(62540B)      SLJ(5STAA)      .                                    03621
2STRD  ZRO(T)           ZRO(T+1)        . FLEX I/O BUFFER                    03622
3STRD  EXF(1000B)       SLJ(N)          .                                    03623
1STWT  SLJ(N)           LDA(2STRD)      . SATELLITE WRITE ROUTINE            03624
       STA(5STBF)       LDA(L-1)        .                                    03625
       ALS(24)          EXF(2000B)      .                                    03626
       SAL(3STWT)       ENA(3STWT)      .                                    03627
       SAU(Z+7)         SIU1(5PXIT)     .                                    03630
       SIL2(5PXIT)      SIU3(5PX22)     .                                    03631
       STQ(5QTMP)       EXF(62560B)     .                                    03632
       SLJ(5STCD)       ZRO(0)          .                                    03633
3STWT  EXF(1000B)       SLJ(N)          . RE-START CLOCK                     03634
1RCVR  SLJ(N)           ENA(77B)        . RECOVERY ROUTINE                   03635
       STA(T)           SLJ4(1STWT)     . WRITE COMPLETION  FLAG TO 160      03636
2RCVR  EXF(21240B)      ENQ(-0)         . RECOVERY                           03637
       LDA(2MSTR)       STA(2M)         .                                    03640
       LDA(3MSTR)       STA(3M)         .                                    03641
       ENA(2A)          SAU(Z+30B)      .                                    03642
       LDA(3RCVR)       STA(7RA)        .                                    03643
       LDA(4RCVR)       STA(7WA)        .                                    03644
       STA(7WB)         STA(7WC)        .                                    03645
       LDA(4KSTR)       STA(4K)         .                                    03646
       EXF(52502B)      EXF(62502B)     . TAKE READ AND WRITE CONTROL        03647
       ENA(11A)         SAL(Z+25B)      . RE-SET MESSAGE HANDLER             03650
       ENA(1SWTH)       SAL(11MA)       .                                    03651
       STQ(R)           STQ(M+39)       .                                    03652
       STQ(0SWT)        SLJ(11MAA)      . RETURN TO 1604 MONITOR JOB         03653
3RCVR  EXF1(T)          EXF7(11B)       .                                    03654
4RCVR  EXF2(T)          EXF7(21B)       .                                    03655

* DD65 SCOPE SIMULATOR FOR AN ON-LINE PRINTER

65DPY  SLJ1(4A)         EXF(77010B)     . SELECT 1604 TO DD65                03656
```

```
        SIU1(65DPX)     SIL2(65DPX)     . SAVE INDEXES                          03657
        SIU3(65PX1)     SIL4(65PX1)     .                                       03660
        SIU5(65PX2)     SIL6(65PX2)     .                                       03661
        STA(DPYAS)      STQ(DPYQS)      . SAVE A/Q                              03662
        EXF7(77010B)    SLJ(4A)         . SENSE DD65 AVAILABLE                  03663
        SSK(DDSWT)      SLJ(4A)         . TEST PRINTER IN RUN/STOP MODE         03664
        ENI1(0)         ENQ(7000B)      .                                       03665
        ENA(20B)        RAD1(ITAB)      .                                       03666
        SSU(ND65R)      STA1(ITAB)      .                                       03667
        ISK1(31)        SLJ(L-2)        .                                       03670
        ENI1(32)        OUT1(ITAB)      .                                       03671
        ENA(400B)       ENI1(32)        .                                       03672
        ENQ(777B)       MEQ1(ITAB)      .                                       03673
        LDA1(ITAB)      STA(BUF65+11B)  .                                       03674
        ENQ(77B)        ENI2(9)         .                                       03675
        ENI3(0)         ENA(0)          .                                       03676
        ENI1(0)         SLJ(65SSU)      .                                       03677
65CYL   INI1(1)         ENI(0)          .                                       03700
65SSU   SSU1(W)         ALS(6)          .                                       03701
        ISK3(5)         SLJ(65CYL)      .                                       03702
        IJP2(L+1)       SLJ(65OUT)      .                                       03703
        ALS(6)          INI1(1)         .                                       03704
        SSU1(W)         STA2(BUF65)     .                                       03705
        ENA(0)          SLJ(65CYL)      .                                       03706
65OUT   LDA(BUF65)      SSU(65XIT)      .                                       03707
        STA(BUF65)      ENI(0)          .                                       03710
        ENI1(10)        OUT1(BUF65)     .                                       03711
65DPX   ENI1(N)         ENI2(N)         . RESTORE INDEXES                       03712
65PX1   ENI3(N)         ENI4(N)         .                                       03713
65PX2   ENI5(N)         ENI6(N)         .                                       03714
        LDA(DPYAS)      LDQ(DPYQS)      . RESTORE A/Q                           03715
65XIT   SLJ(4A)         ZRO(0)          . EXIT                                  03716

        END
        END
```

```
1
 RESIDENT, 0, 4504, 0
 INPUTCDI, 2, 600, 0
 OUTPUTCD, 2, 600, 0
 INPUTBIN, 2, 26, 0
 OUTPUTBN, 2, 27, 0
 FLOATFIX, 0, 13, 0
 FIXFLOAT, 0, 25, 0
 PAUSE   , 0, 23, 0
 STOP    , 0, 6, 0
 ERROREXT, 0, 36, 0
 EFMARK  , 1, 7, 0
 REWINDMT, 1, 7, 0
 BACKSPW , 1, 7, 0
 EFLOFFLO, 2, 177, 0
 EFLOFFIX, 2, 245, 0
 EFIXFFIX, 2, 60, 0
 SINF    , 1, 107, 0
 ASINF   , 1, 145, 0
 COSF    , 1, 113, 0
 ACOSF   , 1, 154, 0
 TANF    , 1, 130, 0
 POLYRT  , 6, 2057, 624
 RSDUMP  , 3, 2737, 3
 JORDAN2 , 3, 257, 0
 GAUSS3  , 5, 501, 0
 MATSUB  , 14, 4560, 4744
 EIGEN   , 4, 3243, 0
 ATANF   , 1, 101, 0
 CTNF    , 1, 146, 0
 ACTNF   , 1, 100, 0
 SECF    , 1, 114, 0
 ASECF   , 1, 141, 0
 CSCF    , 1, 117, 0
 ACSCF   , 1, 134, 0
 SINH    , 1, 113, 0
 ASINH   , 1, 144, 0
 COSH    , 1, 102, 0
 ACOSH   , 1, 163, 0
 TANH    , 1, 76, 0
 ATANH   , 1, 101, 0
 CTNH    , 1, 105, 0
 ACTNH   , 1, 77, 0
 SECH    , 1, 76, 0
 ASECH   , 1, 200, 0
 CSCH    , 1, 115, 0
 ACSCH   , 1, 143, 0
 SQRTF   , 1, 72, 0
 EXPF    , 1, 76, 0
```

```
LOGF     , 1, 103, 0
LOG10F   , 1, 111, 0
FLOATF   , 1, 14, 0
XFIXF    , 1, 25, 0
ABSF     , 1, 5, 0
XABSF    , 1, 5, 0
INTF     , 1, 23, 0
XINTF    , 1, 25, 0
SIGNF    , 2, 11, 0
XSIGNF   , 2, 11, 0
MODF     , 2, 27, 0
XMODF    , 2, 10, 0
DIMF     , 2, 7, 0
XDIMF    , 2, 7, 0
STEP     , 1, 7, 0
XLOCF    , 1, 4, 0
CHAIN    , 2, 5, 0
START    , 0, 2, 0
EXIT     , 0, 2, 0
TIME     , 2, 123, 0
LIMIT    , 1, 17, 0
ALARM    , 1, 30, 0
TROUBLE  , 4, 754, 0
BINDUMP  , 3, 1007, 0
COPYS    , 3, 21, 0
COPY     , 2, 24, 0
VERIFY   , 3, 45, 0
TRANSFER, 3, 47, 0
VFYLIB   , 2, 26, 0
DUMP     , 3, 62, 0
LOAD     , 1, 46, 0
LIST     , 2, 70, 0
LOCATE   , 1, 64, 0
READ     , 4, 125, 0
WRITE    , 4, 141, 0
READMT   , 4, 125, 0
WRITEMT  , 4, 141, 0
TMAP     , 2, 614, 0
LIBFIX   , 3, 466, 0
LIBRARY  , 0, 13, 0
         , 0, 7122, 27122
)N*.* -  , 644, 2076, 76522
         , 752, 42001, 67600
         , 710, 10105, 0
65-4A545, 400, 40673, 42070
+0 .1818, 27, 76512, 30110
-Y4 JY*Y, 72, 56504, 10240
A(S K M#, 203, 20110, 110
*PA55)-P, 40, 14131, 12170
```

```
40A744E4, 547, 30402, 20413
-I4 -+ 8, 11, 110, 70704
TUK  7+2, 50, 53310, 62176
  +9/I +, 641, 10424, 25071
 H+ C64 , 417, 22252, 6105
        , 147, 54075, 31351
4       , 732, 15061, 66003
        , 710, 11575, 0
1 23232S, 50, 14073, 76016
 Q1M    , 100, 100, 1
-ZH1 XC4, 220, 65030, 12070
-Q-RA5 Z, 405, 72230, 2057
-       , 232, 40002, 42311
- S2HAU/, 444, 45457, 26606
B5S2HAXQ, 403, 22027, 26211
# S2I2,1, 74, 14477, 37101
 E+776+5, 630, 40702, 50401
        , 423, 12001, 2225
P C4P AV, 602, 45645, 56033
FORTBIN , 6, 14000, 0
MAPTBIN , 3, 1200, 0
GRAPH   , 4, 4770, 0
GRAPH2  , 17, 4536, 0
ROOTS   , 3, 7720, 0
ROOTS2  , 11, 6414, 0
SORT    , 6, 2013, 0
FORTMAP , 4, 12200, 0
MANTLIB , 7, 15000, 0
FORTRUN , 5, 14000, 0
FORTLIB , 6, 14000, 0
MTTEST  , 0, 62045, 0
COMPSYM , 4, 22751, 0
CONPAC  , 2, 353, 0
REMCON  , 2, 355, 0
```

ENCLOSURES

ENCLOSURE 1

A. IDENTIFICATION

TITLE: 160 Grafplot Routine

PROGRAMMERS: R.L. HOGG / D.C. GLOVER

ORGANIZATION: U.S. Naval Postgraduate School
Monterey, California

DATE: 1 April 1963

B. PURPOSE

To plot multiple curved graphs on the CalComp 165 plotter. Graph titling and axes annotation are provided. The input to the routine is a specially prepared magnetic tape, see the CDC 160-A program of the same name for the input tape format. This routine is a truncated version of the 160-A program, interpolation has been deleted in order to operate in the reduced memory of the 160 computer.

C. USAGE

1. Normal Operation

   a. Load the specially prepared magnetic tape input (See 160-A write-up) on unit #1. Select unit for Binary parity.
   b. Load bi-octal grafplot program at address 0000.
   c. Position the plotter pen at position where the lower left corner of the graph is to be plotted.

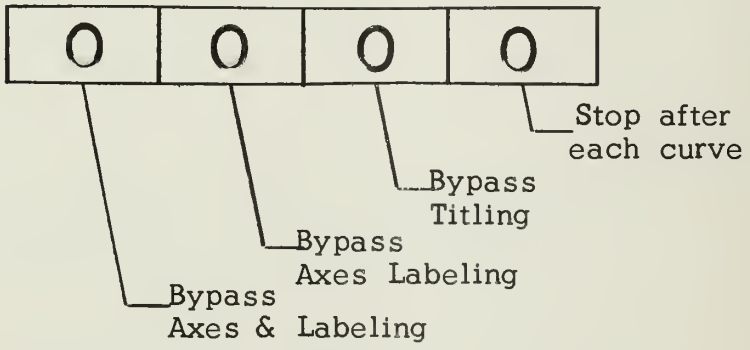
d. Starting Addresses

Location	Contents of "A"	Results
0000	# of graphs to be plotted	Reads tape and plots # of graphs called for. If A=0, all graphs will be processed.
0001		Skip to next graph.
0002		Backspace to beginning of current graph.
0003		Back-up one graph.
0004		Same as a 0000 start.
0010		Rewind tape to Load Pt.
0040	Graph # in A	Skips forward N-1 graphs, into position to plot graph #N.

e. Optional Running Modes

Four modes of operation are provided. To select the mode desired, execute a start at cell 0020 with the following "A" register arguments set. This primes the routine for the mode selected. Remaining graphs will be plotted in this mode, operations as previously described.

"A" REGISTER FORMAT for MODE SELECTION



NOTE: Entry of any non-zero octal number, constitutes setting an argument.

f. Normal Stops

Location	Situation
0004	Normal halt after operation is completed
1157	Optional halt after each curve. Run to continue.

g. Error Stops

0130	*Parity - Title location data
0135	**
0173	*Parity - Title character
0200	**
0373	*Parity - Axed data
0400	**
1125	*Parity - Curve data
1163	**

* Re-Select Start to continue questionable plot.

** Unable to Continue / Skip to next graph to proceed.

A. IDENTIFICATION

TITLE: 160-A Graph Plot Program

DATA CENTERS IDENT: *B001.

PROGRAMMERS: Maurice D. Weir, Milton Spritzer, D. W. McIlhenny, Harry Farnsworth

ORGANIZATION: Fleet Numerical Weather Facility, Monterey, California

DATE: 15 August 1962

REVISED: 4 February 1963, by R. Hogg and D. Glover

B. PURPOSE

To plot N curves on the CONTROL DATA 165 Plotter, complete with titling information and annotated axes, from a specially prepared magnetic tape. SEE TAPE FORMAT.

C. USAGE

1. Computer Operational Procedure

a. Normal operation:

1) Specially prepared magnetic tape on Unit #1. (SEE TAPE FORMAT)

2) Load bi-octal tape at location 0, bank 0. Set direct bank equal to zero.

3) Position the plotter pen such that the cross-hairs line up in the following INITIAL POSITION: LEFT most VERTICAL LINE and one of the HALF-INCH HORIZONTAL LINES on the graph paper.

CAUTION: User should indicate this POSITION in relation to a previously plotted graph. This prevents the forthcoming graph from being written over a previously plotted one.

4) Starting Address

Location	Contents of A	Result
0000	No. of graphs to be plotted.	Reads tape and plots number of graphs called for. If A = 0, tape will be completely processed.

Location	Contents of A	Result
0001		Skip to next file.
0002		Backspace to beginning of current graph.
0003		Back up one complete graph.
0010		Rewind tape to load point.
0040	Graph No. N	Skip N-1 graphs, and stop

NOTE: Normal halt for above operations is location 4.

5) Plotter pen will position itself, draw and annotate the horizontal and vertical axes. Pen will finish by driving back such that the cross-hairs are now lined up with the INTERSECTION of the HORIZONTAL and VERTICAL AXES just drawn.

6) Plotter will plot N curves with their identification (if any). Pen will finish by driving back to the INTERSECTION of the HORIZONTAL and VERTICAL AXES.

7) Normal Stop

(P) = 0004 - Normal halt for all options.

8) Error Stops

a) The following error stops, which occur during the read from the magnetic tape, have two options. These options are:

i) RUN for questionalbe plot, or

ii) SET selective jump switch #2 and RUN. This will result in by-passing that particular section of the program (e.g., titling, annotation of axes, or curve).

The error stops are:
P = 0135 - Parity error in title locations
P = 0176 - Parity error in titling data
P = 0401 - Parity error in axes data
P = 1144 - Parity error in curve data

b) The following error stops, which occur during the read from the magnetic tape, have no options. If any occur, operator should check to make certain magnetic tape unit is on unit #1, tape is at load point, tape is ready, 163-2 units are on. Then set (P) according to step 4).

The error stops in this category are:
P = 0142 - titling location
P = 0203 - titling data
P = 0406 - Axes data
P = 1200 - curve data

P = 1103 - error in number of interpolations. RUN.

b. Operational variations:

The Graph Plot program contains some special operational procedures which allow for variations in the graph by altering the Normal operation.

1) Option to by-pass titling:

After the plotter pen has been positioned in the previously defined INITIAL POSITION, SET selective jump switch #2. Set (P) according to step a.4) and RUN. Titles will be read from magnetic tape but not plotted. Program will make NORMAL STOP P = 0361, and pen will not have moved.

2) Option to by-pass axes annotation:

With plotter pen at AXES INTERSECTION and at SPECIAL STOP P = 0361, SET selective jump switch #2 and RUN (do not clear). Axes data will be read from magnetic tape but not plotted. Program will make NORMAL STOP P = 1051, and pen will not have moved.

3) Option to PLOT - one-curve-STOP:

With plotter pen at AXES INTERSECTION and at SPECIAL STOP P = 1051, SET selective stop switch #1 and RUN. After curve is plotted, pen will drive back to AXES INTERSECTION, and program will make SPECIAL STOP P = 1174 before the next curve record is read in. Pen-ink may now be changed for next curve, etc. RUN to read in and plot next curve.

4) Option to by-pass one curve at a time:

1136

At SPECIAL STOP P = 1051, SET selective jump switch #2 and RUN. Curve data will be read in, and program will make SPECIAL STOP P = ~~1065~~. If selective jump switch # 2 remains set when RUN is set, next curve will be read in, and program will again make SPECIAL STOP P = ~~1065~~. If selective jump switch # 2 is not set when RUN is set, next curve will be read in and plotted, etc.

2. Space Requirements

a. Program: 0 - 4137 in Bank zero.

b. Curve data and interpolation: 100 - 7777 in Bank one.

3. Temporary Storage

Locations 50 - 77 in Bank zero.

4. Tape Mountings

Magnetic tape must be read from unit # 1.

5. Caution to Users

Do not step through program. Read write-up carefully before using the Graph Plot Program.

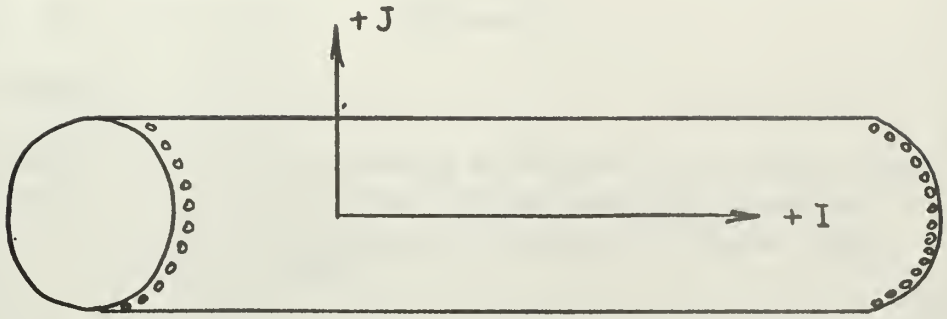
D. TAPE FORMAT

Each magnetic tape to be used by this program must have one or more records (as indicated) in each of the following categories:

I	Titling data	- multiple records
II	Axes data	- one record
III	Curve data	- multiple records
IV	End of File.	

These records must be on the magnetic tape in the order defined above and in the format to be specified below.

AXES DEFINED AS FOLLOWS:



I = horizontal axis

J = vertical axis

I Titling data

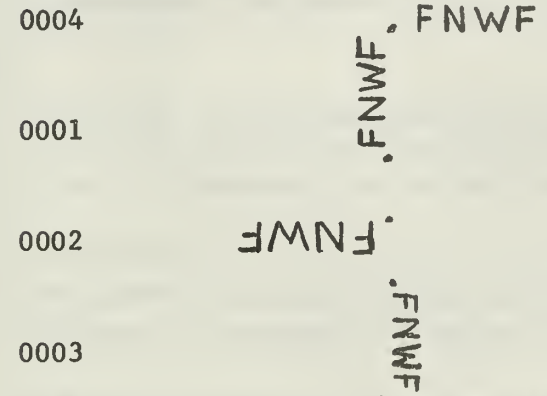
1. Record 0

a. 12 bit binary mode

b. 4 160-A words (i.e., 1 1604 word)

c. **Format:**

word 0 (I_T)	I distance of initial pt. of first title from the INITIAL POSITION, given as a signed Δ I movement in inches times 100_d.
word 1 (J_T)	J distance of initial pt. of first title from the INITIAL POSITION, given as a signed Δ J movement in inches times 100_d.
word 2	Orientation of titles defined as follows:



word 3	Number of lines in title given as a 12 bit integer. (OCTAL)

2. Records 1 to T

a. BCD characters in 6-bit binary mode.

b. Maximum length of 120_d characters where 119_d is number of characters to be plotted.

c. **Format:**

character 0	Size of letters in title line. Size X is defined to be "X times one-tenth inch". The width of a letter is approximately equal to its height.

character 1 : : character 119	Title line in BCD. Not necessary to fill out this line

II Axes data

a. 1 record in 12 bit binary mode

b. 20_d 160-A words

c. Format:

word 0 (I_H)	I distance of initial point of horizontal axis from the INITIAL POSITION, given as a signed ΔI movement in inches times 100_d.
word 1 (J_H)	J distance of initial point of horizontal axis from the INITIAL POSITION, given as a signed ΔJ movement in inches times 100_d.
word 2 (L_H)	Length of the horizontal axis given as a signed ΔI movement from where the axis begins in inches times 100_d.
word 3 (I_{HL})	I distance of initial point of first horizontal axis label from the INITIAL POSITION, given as a signed ΔI movement in inches times 100_d.
word 4 (J_{HL})	J distance of initial point of first horizontal axis label from the INITIAL POSITION, given as a signed ΔJ movement in inches times 100_d.
word 5 (I_{HL})	I distance between horizontal labels, given as a signed ΔI movement in inches times 100_d.
word 6 (V_H)	First value which is to be plotted for the horizontal axis labeling. This is a signed 12 bit integer which will be converted to BCD. Its maximum value cannot exceed $\pm 999_d$.
word 7 (V_H)	Value of the ΔI distance between horizontal axis labels given as a signed 12 bit integer.
word 8 (N_H)	Number of horizontal axis labels to be plotted. This is always a positive 12 bit integer.

word 9 (S_H)

Upper six bits: Size of letters in horizontal axis labeling.

(O_H)

Lower six bits: Orientation of horizontal axis labels. Orientation previously defined.

word 10 (I_V)

I distance of initial point of vertical axis from the INITIAL POSITION, given as a signed ΔI movement in inches times 100_d.

word 11 (J_V)

Vertical axis analogy to word 1.

word 12 (L_V)

Vertical axis analogy to word 2.

word 13 (I_{VL})

Vertical axis analogy to word 3.

word 14 (J_{VL})

Vertical axis analogy to word 4

word 15 (J_{VL})

Vertical axis analogy to word 5

word 16 (V_V)

Vertical axis analogy to word 6.

word 17 (V_V)

Vertical axis analogy to word 7.

word 18 (N_V)

Vertical axis analogy to word 8.

word 19 (S_V)

Upper six bits: Size of letters in vertical axis labeling.

(O_V)

Lower six bits: Orientation of vertical axis labels.

III Curve data

a. Multiple records in 12 bit binary mode.

b. m words long where m has maximum value of 3677_8.

c. Each I, J must be given as a distance from the INITIAL POSITION, as a signed $\Delta I, \Delta J$ movement in inches times 100_d.

d. Format:

word 0	0000
word 1	0000
word 2	0000
word 3	number of interpolations (00 - 07)
word 4	I_1
word 5	J_1
word 6	I_2
word 7	J_2
⋮	
word m-5	3777 end flag
word m-4	3777 end flag
word m-3	curve name. 4 BCD characters,
word m-2	packed two to a word.
word m-1	upper six bits: letter size of curve name
	lower six bits: orientation of curve name

IV End of File

E. TIMING

Each graph takes 2 - 3 minutes to plot, including titling, annotated axes, and interpolation.

F. METHOD

1. General

All location points on the graph are interpreted as being relative to the (0, 0) INITIAL POSITION previously defined. All lengths, intervals, etc. must have the appropriate sign according to the definition of the(I, J) axes. Information is plotted as follows: titles, horizontal axis, horizontal axis annotation, vertical axis, vertical axis annotation, and finally the curve(s) (with interpolations, if any). The program uses TAP for reading the magnetic tape input, OVDR for driving to a position with the pen up, LINE for joining given points with the pen down, and LABEL for titling.

2. Titles

The initial points from the magnetic tape input are used to position the pen in the desired location on the graph for the first title. The

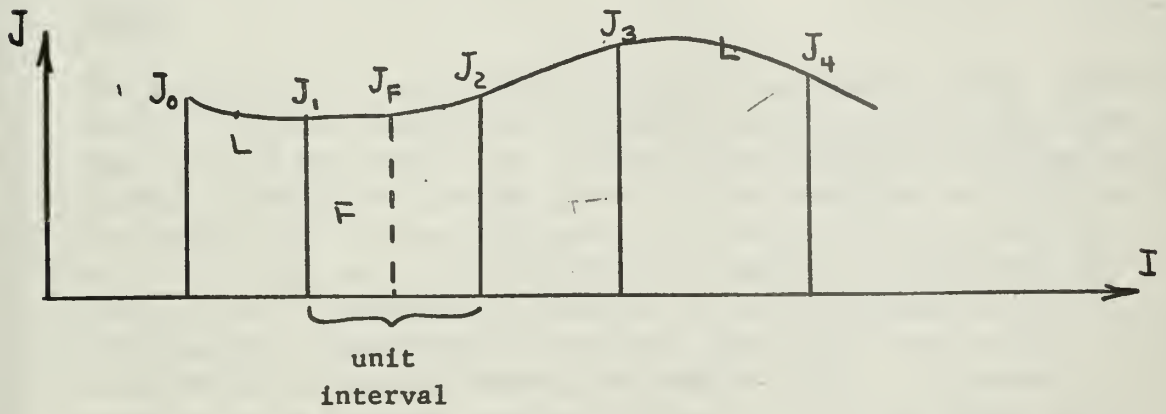
first title is plotted, and if there are additional titles, the program calculates the appropriate spacing between lines according to the orientation and the letter size of the incoming title. The program keeps track of all pen movements so that when the last title is plotted, pen drives back to the (0, 0) INITIAL POSITION.

3. Annotation of Axes

The initial points from the magnetic tape input are used to position the pen in the desired location on the graph for the start of the horizontal axis. The axis is drawn, and then the pen moves into position to begin the first horizontal axis label. This first label is plotted according to its size and orientation; the remaining labels being computed and then plotted at the desired intervals, based upon the initial value, the scaling, and the number of labels desired. At the completion of the horizontal axis annotation, the pen positions itself at the start of the vertical axis. The axis is drawn and labelled as in the horizontal axis annotation. The program keeps track of all pen movements so that when the vertical axis annotation has been completed, pen drives back to the INTERSECTION of the HORIZONTAL and VERTICAL AXES.

4. Interpolation

This is a four-point interpolation method based on the Bessel Interpolation Equations.



The interpolation scheme fits the four points to a quadratic curve, computing the interpolated point J_F by the following equation:

$$J_F = J_1 + F\,(J_2 - J_1) + \frac{F\,(F - 1)}{4}\,(J_0 + J_3 - J_2 - J_1)$$

The program uses F = 1/2.

This interpolation method depends upon equal I intervals. Since it cannot perform an interpolation between the first and last pairs of J values, a linear interpolation is performed at these two positions (marked "L" in the diagram). This linear interpolation occurs at the half-interval of both end pairs, which allows the user to make more than one interpolation by maintaining equality of I intervals.

The number of interpolations X desired must comply with the following inequality:

$$\frac{\Delta I}{2^X} > \frac{1}{100}$$

The program has set the maximum number of interpolations as 07. If the number of interpolations indicated (see TAPE FORMAT, "curve data") exceeds this maximum, program will interpolate 07 times only, and indicate an error in the number of interpolations indicated.

Caution: Because of the necessary storage space for the curve data points, the maximum number of (I, J) points that can be plotted after interpolation is 980_d.

WARNING: Since the interpolation depends upon equal spacing of the I intervals, any curve which does not have equally spaced points will not be properly interpolated. This applies to multiple-valued functions and asymptotic values.

5. Curves

The first (I, J) curve data point from the magnetic tape input is used to position the pen in the desired location on the graph for the start of the curve plot. Curve is then plotted by joining successive points by a straight line. At the completion of plotting the curve (with the desired number of interpolations, if any), program labels curve (if desired) with indicated orientation and letter size. Program keeps track of all pen movements so that when the curve labelling is completed, pen drives back to the INTERSECTION of the HORIZONTAL and VERTICAL AXES. If another curve is to be plotted, program proceeds in the same manner as for the first curve. When the final curve is plotted, pen will be positioned at the INTERSECTION of the HORIZONTAL and VERTICAL AXES.

Caution: User may not desire a curve label, in which case blanks are indicated on the magnetic tape for the four BCD characters.

(see TAPE FORMAT, "curve data.") Nevertheless, an orientation and letter size must still be indicated since they are required by the LABEL subroutine.

FINIS

A. IDENTIFICATION

TITLE: 1604 Graph Plotting Subroutine

DATA CENTERS IDENT: *A002.

PROGRAMMERS: R. L. Hogg and D. C. Glover

ORGANIZATION: Naval Postgraduate School, Monterey, California

DESCRIPTION WRITE-UP: E. A. Campbell, CDC

DATE: 5 March 1963

B. PURPOSE

To write data on a magnetic tape for off-line plotting on the CONTROL DATA 165 plotter. The tape format is compatible with the 160-A Graph Plot Program (Data Centers Ident *B001.) which is used to drive the 165 plotter.

C. USAGE

1. Calling Sequence

a. FORTRAN-62

CALL GRAFPLOT(MODCURV, NUMPTS, X, Y, LABEL, INTERPS, SFX, SFY, IXOFFS, IYOFFS, IWIDE, IHIGH, MODE, TITLESIZ, TITLE1, TITLE2, ·····, TITLEN)

CALL AUTOGRAF(MODCURV, NUMPTS, X, Y, TITLE1, TITLE2, ·····, TITLEN)

b. CODAP1

Either:

```
        ENA     (MODCURV)
        ENQ     (NUMPTS)
        RTJ     GRAFPLOT
    +   ZRO     X
        ZRO     Y
        ZRO     LABEL
        ZRO     INTERPS
        ZRO     SFX
        ZRO     SFY
        ZRO     IXOFFS
        ZRO     IYOFFS
        ZRO     IWIDE
        ZRO     IHIGH
        ZRO     MODE
        ZRO     TITLESIZ
        ZRO     TITLE1
        ZRO     TITLE2
         :
        ZRO     TITLEN
```

```
Or:         ENA     (MODCURV)
            ENQ     (NUMPTS)
            RTJ     AUTOGRAF
      +     ZRO     X
            ZRO     Y
            ZRO     TITLE1
             :
             :
            ZRO     TITLEN
      +     RTJ     ERROR
```

2. Input Parameters

Parameter Name	Value	Description
*MODCURV	0	First and last curve for this graph
	1	First curve of many
	2	Intermediate curve
	3	Final curve for this graph
NUMPTS	1-900	Number of points
X		Location of X coordinates
Y		Location of Y coordinates
LABEL	4 BCD Char.	BCD label for curve
	0	No label
	-0	Automatic sequencing of curves for No. 1-8.
INTERPS	0	No interpolation between coordinate points
	1-7	One to 7 interpolations between coordinate points
SFX	0	Program will find best scale factor
	Fl. Pt. Number	Scale factor for X **
SFY	0	Program will find best scale factor
	Fl. Pt. Number	Scale factor for Y **
IXOFFS	1-10	Number of inches to offset X axis
	0	No offset
IYOFFS	1-10	Number of inches to offset Y axis
	0	No offset
IWIDE	1-10	Length of X axis - inches
IHIGH	1-120	Length of Y axis - inches

*If MODCURV is 2 or 3, then only the first 6 parameters need to be repeated.

**Scale factor = 100/units per inch on plotted output.

Parameter Name	Value	Description
MODE	0	Automatically calculate best axes offset for origin at lower left
	1	Locate origin at center
	2	Locate origin at lower left
	3	Locate origin at upper left
	4	Locate origin at upper right
	5	Locate origin at lower right
TITLESIZ	1-24	Multiplier for lettering size. Basic size is .1 inches.
TITLE1		Location of first title. Title array must be at least 10 words of BCD characters. Fill unused characters with blanks.
⋮		
TITLEN		Location of Nth title array

3. Space Requirements

1120 decimal locations

6. Error Returns

No error return is used, however, an error return point is skipped in order to be compatible with FORTRAN subroutines.

8. Input/Output Tape Mountings

Plot data output on logical 48 through CO-OP Monitor.

9. Input/Output Tape Formats

See 160-A Graph Plot Program write-up for details. The basic form is titling information, axes annotation, and curve data with an end of file separating graphs. No special end of tape mark is used to terminate the tape.

13. Cautions to Users

a. If scale factors are defined, then the appropriate axes offsets should also be defined.

b. The normal width of the plotter paper is 10 inches. For best results, do not use a horizontal axis length greater than 9 inches.

c. The number of interpolations N required must comply with the following inequality: $\Delta I/2^n > 1/100$ where ΔI is the uniform (equal) increment along the horizontal axis. See reference for further details.

14. Equipment Configuration

CO-OP Monitor configuration with at least one output tape available.

15. References

This program uses 160-A Graph Plot Program (Data Centers Ident No. *B001) as a subset to drive the 165 plotter.